Software Piracy and Outsourcing in Two-Sided Markets[*]

Masakazu Ishihara
Stern School of Business
New York University

Eitan Muller
Stern School of Business
New York University
Arison School of Business
Interdisciplinary Center (IDC) Herzliya

May 10, 2018

Abstract

This paper examines the role of software piracy in digital platforms, where a platform provider makes a decision of how much software to produce in-house and how much to outsource from a third-party software provider. Using a vertical differentiation model, we first theoretically investigate the impact of software piracy on equilibrium pricing and profits of the platform and software providers, and software outsourcing decision by the platform. We find that the platform provider can benefit from piracy, and that an increase in piracy reduces in-house software production. We then provide empirical evidence for the external validity of our theoretical prediction on the outsourcing decision using data from the U.S. handheld video game market between 2004 and 2012. This market is a classical two-sided market, dominated by two handheld platforms (Nintendo DS and Sony PlayStation Portable) and is known to have suffered from software piracy significantly. Our regression results show that the proportion of in-house software decreases in piracy, supporting our theoretical prediction.

JEL Classification: D21, D22, K42, L24, L86

Keywords: Software Piracy, Two-Sided Markets, Outsourcing, Video Games

[*] We thank Marc Rysman and Steven Shugan for their valuable comments. We also thank seminar and conference participants at U of Maryland, NYU, Stanford, UCSD, International Industrial Organization Conference, Marketing in Israel Conference, and Marketing Science Conference.

1 Introduction

Software piracy has been a hotly debated topic in digital platforms such as video games, smartphone/tablet apps, and ebooks. Traditionally, studies on software piracy have mainly focused on how software piracy might increase/decrease the profits of software providers (see e.g., Conner and Rumelt 1991, Takeyama 1994, Givon, Mahajan, and Muller 1995, Shy and Thisse 1999, Peitz 2004, Jain 2008, Sinha, Machado, and Sellman 2010, Vernik, Purohit, and Desai 2011, Lahiri and Dey 2013). However, in digital platforms where consumers and software providers interact (e.g., Church and Gandal 1992), software piracy does not only affect the profits of software providers, but also the profits of platform providers. In order to use software, consumers first need to adopt a platform. This feature appears to suggest that platform providers might benefit from software piracy because it potentially increases the sales of platforms, which creates a conflict of interest in piracy protection between platform providers and software providers.

Despite its importance and relevance to digital platform businesses, little has been studied about the role of software piracy in a two-sided market setting. Notable exceptions are Rasch and Wenzel (2013, 2015). Built on the literature on two-sided markets (e.g., Rochet and Tirole 2006, Rysman 2009), Rasch and Wenzel (2013) theoretically study the conflict between platforms and software developers in a competitive platform market. Rasch and Wenzel (2015) extend this theoretical model and examine how the impact of piracy differs across prominent and non-prominent software developers. Our paper aims at contributing to this literature by examining the role of outsourcing decisions by a platform provider when software piracy exists. In many digital platforms, platform providers are also software providers (e.g., Nintendo, Microsoft, Apple, Google), and often in-house software accounts for a significant proportion of platform providers' profits. For example, for Nintendo DS, a handheld video game device released by Nintendo in November 2004, Nintendo made USD 89.2 million revenue in the first year from own in-house software alone, and this was 53% of revenues for all software released on Nintendo DS and 25% of the revenue from Nintendo DS handheld device (hardware) in the same period. The in-house software production in digital platform markets is an important phenomenon, and has been studied in the context of vertical integration between platform providers and software providers (e.g., Lee 2013). However, no prior studies on software piracy have incorporated this important aspect into the analysis.[1]

When software is provided by both platform and software providers, the effect of piracy on

[1] For example, Rasch and Wenzel (2013, 2015) assume that all software is provided by independent software providers. However, they model platform competition, which we do not study in this paper.

platform providers is not straightforward. While piracy might help increase the sales of platforms, it can hurt in-house software profits. Platforms might then pass on the loss to software providers by outsoucing software, but doing so will reduce the overall profits from software because the margin from in-house software is higher than that from license fee revenues. Our goal is to examine the impact of software piracy on the equilibrium outsourcing decision, or equivalently in-house production decision.

To investigate the question, we develop a vertical differentiation model of software piracy where upon buying a platform, consumers choose to buy a legal copy of software or use an illegal copy. Following previous studies, we capture the degree of piracy through the deteriorated quality of illegal software (i.e., psychological disutility, cost of acquiring knowledge for pirating software). Under this setting, we first consider two baseline scenarios: (1) full integration scenario, where the platform provider supplies both platform and software, and (2) full outsourcing scenario, where the platform provider supplies platform and the software provider supplies software. We show that in the full integration scenario, the platform provider's combined profits from hardware and software are always decreasing as the degree of software piracy increases. While the hardware profits are increasing in piracy, the loss of software profits due to piracy outweighs the gain in hardware profits. However, if the platform provider fully outsources software and earns software license fees from the software provider, the impact of software piracy on its profits is non-linear in piracy, and the platform provider benefits from an increase in piracy when the degree of piracy is relatively high. This is because, although the profits from software license fees are decreasing in piracy, the rate of the decline is smaller than that in in-house software profits in scenario (1). As a result, the gain in platform profits due to piracy can outweigh the loss in license profits when the degree of piracy is relatively high. We also find that under some conditions, the equilibrium license fee is negative.

We then examine the main scenario in which the platform provider chooses the degree of in-house versus outsourced software production. Once again, we find that the platform provider gains from an increase in piracy when the degree of piracy is high. Moreover, as the degree of piracy increases, the optimal proportion of in-house software decreases. This can be explained through the mechanism combined from the two baseline scenarios: As the degree of piracy increases, the loss in in-house software profit due to piracy increases. Although the profit margin from in-house software is higher than license fees, the platform provider benefits from shifting software profits from in-house to license fees because the (negative) marginal impact of piracy can be reduced.

To provide empirical evidence for the external validity of our theoretical result on equilibrium outsourcing decision, we collect data from the U.S. handheld video game market between 2004 and 2012. This market is a classical two-sided market (Clements and Ohashi 2005, Dubé, Hitsch, and Chintagunta 2010, Chao and Derdenger 2013, Derdenger and Kumar 2013, Lee 2013, Derdenger 2014), dominated by two handheld platforms (Nintendo DS and Sony PlayStation Portable) and is known to have suffered from software piracy significantly (Fukugawa 2011). For Nintendo DS, a device called Revolution 4 made hacking possible, and for Sony PlayStation Portable, Pandora battery was the key device for hacking. We obtain monthly data on software sales from NPD and create two measures for the proportion of in-house software based on (i) the number of in-house versus outsourced software titles and (ii) revenues of in-house versus outsourced software. As we cannot directly observe the degree of piracy, we use U.S. Google Trends search volume on the two idiosyncratic devices as a proxy for the degree of piracy. Our identification assumption is that as more information about hacking devices becomes available, the chance of a consumer finding hacking information becomes higher, which induces more search behavior on Google. Under this assumption, U.S. Google Trends search volume captures variation in the cost of acquiring knowledge about pirating software. In order to control for potential endogeneity of the search volume, we also obtain Google Trends data on the same devices restricted to Japan (in Japanese).

Using monthly observations for the two handheld devices, we run regression analyses and estimate the effect of piracy on the proportion of in-house software by controlling for other variables such as the cumulative sales of hardware, system software updates, and platform- and month-fixed effects. We find that for both measures of the proportion of in-house software, the effect of piracy was negative and significant. These results support the theoretical prediction that the proportion of in-house software decreases as the degree of piracy increases.

2 A Model of Software Piracy

We examine the role of piracy in outsourcing software in a vertical differentiation model with a monopolist platform provider.[2] Our model consists of three players: the platform provider, the software provider, and consumers. The platform provider produces the hardware and sets the hardware price. It can also produce software by itself (in-house software), outsource software to the software provider (outsourced software), or mix of them. We assume that the total software developed is one unit, and let $\delta \in [0, 1]$ be the proportion of software developed in-house and $1 - \delta$

[2] Throughout the paper, we use hardware and platform interchangeably.

be the proportion outsourced. We assume that the cost of developing δ software for the platform provider is $\frac{C_h}{2}\delta^2$ and the cost of developing $1-\delta$ software for the software provider is $\frac{C_s}{2}(1-\delta)^2$. Throughout the analysis, we assume that $C_h \geq C_s > 0$. Finally, we assume that software is undifferentiated, and that the marginal costs of hardware and software are zero.

The timeline of the model is as follows:

1. The platform provider sets the price of hardware (p_h) and the proportion of software developed in-house (δ).
    - If $\delta < 1$ (some outsourcing), the platform provider sets the unit licensing fee (f) paid by the software provider.
    - If $\delta = 1$ (no outsourcing), the platform provider sets the price of software (p_s).
2. If $\delta < 1$, the software provider observes f and sets the price of software (p_s) that is common for all software.
3. Consumers observe p_h and p_s, and decide to buy hardware and to buy or pirate software.

We first describe consumers' purchase decisions, and then move on to the firms' decisions.

2.1 Consumers' purchase decisions

Suppose that the consumers who buy one unit of software have the following utility:

$$u_{legal}(\alpha) = v - p_h + \alpha - p_s, \tag{1}$$

and the pirates have the following utility:

$$u_{pirate}(\alpha) = v - p_h + \gamma\alpha, \tag{2}$$

where $v > 0$ is the benefit from the hardware absent software; For example, smart phones have some benefit even without any apps, or PlayStation 4 has a built-in Blu-ray player;[3] α is the benefit from the software; p_h is the price of the hardware; p_s is the price of the software; γ is the reduction in utility due to the fact that the software is pirated. It might be psychological disutility such as the fear of getting caught, the cost of acquiring knowledge for finding and using pirated software,

[3] Two platforms in our empirical application, Nintendo DS and Sony PlayStation Portable, have this feature. For example, the hardware benefit for Nintendo DS may come from pre-existing software (Rasch and Wenzel 2013) as Nintendo DS is backward-compatible with Game Boy Advance cartridges. Sony PlayStation Portable has a built-in media player that can can play music and video, and an internet web browser.

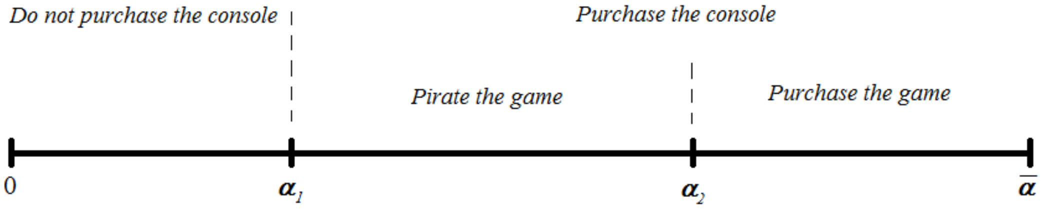


Figure 1: Distribution of buyers and pirates

the fact that the software is not pirated right away and so it's a somewhat older game, or the fact that many pirated software have limited functionality, e.g., inability to play multiplayer sessions online for video games. Thus γ measures the effect of piracy. The larger it is, the more serious is the problem they pose. If γ =1, then no one buys the software, and if $\gamma = 0$, there's no piracy. Thus γ denotes the degree of piracy, and we assume that $\gamma \in (0, 1)$.

We assume that in order to play the software one has to purchase the hardware, that is, the reality of the digital platform business is such that the hardware cannot be pirated. In order to figure out who pirates and who buys, we let α be uniformly distributed between 0 and $\bar{\alpha}$, and as we show shortly, Figure 1 summarizes the distribution of buyers and pirates: To show that indeed consumers with $\alpha \in [0, \alpha_1]$ do not buy the hardware, consumers with $\alpha \in [\alpha_1, \alpha_2]$ buy the hardware and pirate the software, while consumers with $\alpha \in [\alpha_2, \bar{\alpha}]$ buy both hardware and software, note that the utilities of both buyers and pirates are increasing in α and therefore if we define α_1 as the lowest benefit such that $u_{pirate}(\alpha_1) = 0$, then for $\alpha < \alpha_1$, the consumers do not purchase the hardware and thus are out of the market, and for $\alpha > \alpha_1$, the consumers buy the hardware and have only to decide whether to pirate or purchase the software. Solving $u_{pirate}(\alpha_1)$ from Equation (2), this lower bound is given by the following:

$$\alpha_1 = \frac{p_h - v}{\gamma}. \tag{3}$$

The boundary α_2 is such that the utilities of the pirates and legal buyer are exactly equal. Since from Equations (1) and (2), it is evident that $\frac{\partial u_{legal}}{\partial \alpha} > \frac{\partial u_{pirate}}{\partial \alpha}$, consumers with $\alpha < \alpha_2$ pirate the software and those with $\alpha > \alpha_2$ buy it legally. Solving $u_{pirate}(\alpha_2) = u_{legal}(\alpha_2)$ from Equations (1) and (2) yields the following:

$$\alpha_2 = \frac{p_s}{1 - \gamma}. \tag{4}$$

We now discuss the optimal behavior of the firms where we first deal with one firm that produces both hardware and software ($\delta = 1$: full integration), and then proceed to the case where production is done by separate entities ($\delta = 0$: full outsourcing). Finally, we consider the case where $\delta \in (0, 1)$. Throughout our analysis, we make the following assumption.

Assumption 2.1. *$\bar{\alpha} > \sqrt{2}v$.*

Intuitively, this assumption states that the utility from software is large enough as compared to the pure hardware benefit. We make this assumption because our focus in this paper is on the role of software and its piracy activities.

2.2 A single firm produces both hardware and software (Full Integration)

The profit of the monopolist producing both hardware and software is given by the following:

$$\pi_h = p_h(\bar{\alpha} - \alpha_1) + p_s(\bar{\alpha} - \alpha_2) - \frac{C_h}{2} = p_h\left(\bar{\alpha} - \frac{p_h - v}{\gamma}\right) + p_s\left(\bar{\alpha} - \frac{p_s}{1-\gamma}\right) - \frac{C_h}{2} \tag{5}$$

where the RHS of Equation (5) is achieved by substituting from Equations (3) and (4). First order conditions with respect to both prices yield the following (it is straightforward to check the second order conditions are also satisfied). The superscript I implies that the variable in question is an equilibrium solution in this Integration case.

$$p_s^I = \frac{\bar{\alpha}(1-\gamma)}{2}, \tag{6}$$

$$p_h^I = \frac{v + \bar{\alpha}\gamma}{2}. \tag{7}$$

Substituting Equations (6) and (7) into (3) and (4) yield the following:

$$\alpha_1^I = \frac{\bar{\alpha}\gamma - v}{2\gamma}, \tag{8}$$

$$\alpha_2^I = \frac{\bar{\alpha}}{2}. \tag{9}$$

Clearly $\alpha_2^I > \alpha_1^I$ for all range of parameters, and $\alpha_1^I \geq 0$ if $\bar{\alpha}\gamma \geq v$. Assumption 2.1 guarantees that such γ exists. If this condition is not satisfied, it means that all consumers purchase the hardware and the hardware price does not depend on γ (i.e., $p_h = v$). It is also easy to check that $u_{pirate}(\alpha_2) = u_{legal}(\alpha_2) = \frac{v}{2}$. Thus the utility of the pirates spans the range from 0 to $\frac{v}{2}$, while the utility of the legal buyers span the range of $\frac{v}{2}$ to $\frac{\bar{\alpha}+v}{2}$. Substituting Equation (6) through (9) into (5) yields that the profits of the firm are declining with increase in piracy (γ) as is evident from the following equation:

$$\pi_h^I = \begin{cases} \bar{\alpha}v + \frac{\bar{\alpha}^2(1-\gamma)}{4} - \frac{C_h}{2} & \text{if } \gamma < \frac{v}{\bar{\alpha}}, \\ \frac{(\bar{\alpha}\gamma+v)^2}{4\gamma} + \frac{\bar{\alpha}^2(1-\gamma)}{4} - \frac{C_h}{2} & \text{if } \gamma \geq \frac{v}{\bar{\alpha}}. \end{cases} \tag{10}$$

Equation (10) also suggests that when C_h is large, the hardware firm prefers not to produce software at all. If the firm does not produce any software, it still can make profits on the hardware because of the intrinsic value of hardware v. The firm would then charge the price of $p_h = v$, and at that price the entire market ($\bar{\alpha}$) would buy the hardware. Thus the firm would make the profits of $\bar{\alpha}v$.

$$\pi_h = \bar{\alpha}v.$$

It is then easy to check that when C_h is large, a higher γ makes the hardware firm not produce software because $\frac{\partial \pi_h^I}{\partial \gamma} < 0$. The following Proposition summarizes the results of the Full Integration case.

Proposition 2.2. *[Full Integration] For small C_h, the hardware firm will develop software for any $\gamma \in (0, 1)$. For intermediate C_h, the hardware firm will only develop software when γ is smaller than a threshold that is a function of $(\bar{\alpha}, v, C_h)$. For large C_h, the hardware firm will only sell hardware. When the hardware firm develops software, (i) the firm's profits decrease in γ and (ii) for a small γ, all consumers buy hardware at $p_h = v$ and the profits from hardware sales do not depend on γ.*

We provide detailed analysis in Appendix A.1.

2.3 Separate production with hardware firm setting licensing fee (Full Outsourcing)

This case deals with two independent firms: The hardware firm (subscripted by h) sets the price of the hardware (p_h) and the licensing fee (f), and the software provider (subscripted by s) sets the price of the software (p_s). We start with the software provider's problem.

The profit function of the software provider is given by the following, where the RHS of the equation is achieved by substituting from Equation (4). In accordance with the previous section, the superscript O implies that the variable in question pertains to this Outsourcing case.

$$\pi_s = (p_s - f)(\bar{\alpha} - \alpha_2) - \frac{C_s}{2} = (p_s - f)\left(\bar{\alpha} - \frac{p_s}{1-\gamma}\right) - \frac{C_s}{2}. \tag{11}$$

The first-order condition gives

$$p_s^O = \frac{\bar{\alpha}(1-\gamma)}{2} + \frac{f}{2}. \tag{12}$$

Substituting Equation (12) into Equation (11) yields the software provider's profits:

$$\pi_s(f) = \frac{(\bar{\alpha}(1-\gamma) - f)^2}{4(1-\gamma)} - \frac{C_s}{2}. \tag{13}$$

This profits impose an important constraint: the software provider produces the software only if $\pi_s(f) \geq 0$. Since $\frac{\partial \pi_s(f)}{\partial f} < 0$, when C_s is large, the hardware firm needs to lower the licensing fee sufficiently (f could be negative) in order to make the software provider produce the software.

The hardware producer profits are given by the following where the RHS of the equation is achieved by substituting from Equations (3), (4) and (12).

$$\pi_h = p_h(\bar{\alpha} - \alpha_1) + f(\bar{\alpha} - \alpha_2) = p_h \left(\bar{\alpha} - \frac{p_h - v}{\gamma} \right) + f \left(\bar{\alpha} - \frac{\bar{\alpha}(1-\gamma) + f}{2(1-\gamma)} \right). \tag{14}$$

The first-order conditions with respect to p_h and f yield the following:

$$p_h^O = \frac{v + \bar{\alpha}\gamma}{2}, \tag{15}$$

$$f^O = \frac{\bar{\alpha}(1-\gamma)}{2}. \tag{16}$$

If we substitute Equation (16) into (12), we get the software price:

$$p_s^O = \frac{3\bar{\alpha}(1-\gamma)}{4}. \tag{17}$$

The price of the hardware remains as before, but the price of the software increased, and thus we see the effect of double marginalization: the price to the consumers is higher. Also we can recalculate the boundaries by substituting Equations (15) and (17) into (3) and (4):

$$\alpha_1^O = \frac{\bar{\alpha}\gamma - v}{2\gamma}, \tag{18}$$

$$\alpha_2^O = \frac{3\bar{\alpha}}{4}. \tag{19}$$

Thus the lower bound α_1 did not change while the upper bound increased from $\frac{\bar{\alpha}}{2}$ to $\frac{3\bar{\alpha}}{4}$. Thus the fact that a separate firm produces the software, increases pirates (since it increases the software price). We can now compute the profits of the two firms as follows:

$$\pi_h^O = \begin{cases} \bar{\alpha}v + \frac{\bar{\alpha}^2(1-\gamma)}{8} & \text{if } \gamma < \frac{v}{\bar{\alpha}}, \\ \frac{(\bar{\alpha}\gamma + v)^2}{4\gamma} + \frac{\bar{\alpha}^2(1-\gamma)}{8} & \text{if } \gamma \geq \frac{v}{\bar{\alpha}}, \end{cases} \tag{20}$$

$$\pi_s^O = \frac{\bar{\alpha}^2(1-\gamma)}{16} - \frac{C_s}{2}. \tag{21}$$

Now we can ask about the effect of piracy (γ). Suppose γ is increased. Then the price of the software drops while the hardware price increases. In the Full Integration case, the absolute changes are equal, i.e., $\left|\frac{\partial p_h}{\partial \gamma}\right| = \left|\frac{\partial p_s}{\partial \gamma}\right| = \frac{\bar{\alpha}}{2}$. In the Full Outsourcing case, the decrease in software price is higher than the increase in the price of the hardware as demonstrated by the following inequality:

$$\left|\frac{\partial p_h}{\partial \gamma}\right| = \frac{\bar{\alpha}}{2} < \frac{3\bar{\alpha}}{4} = \left|\frac{\partial p_s}{\partial \gamma}\right|.$$

More interestingly, while the software firm clearly loses from piracy (see Equation 21), the hardware firm may benefit from piracy when $\gamma \geq \frac{v}{\bar{\alpha}}$. Differentiating Equation (20) with respect to γ yields that $\frac{\partial \pi_h}{\partial \gamma} \geq 0$ if:

$$\bar{\alpha}\gamma \geq \sqrt{2}v. \tag{22}$$

If $v \leq \bar{\alpha}\gamma \leq \sqrt{2}v$, then we get the expected result that $\frac{\partial \pi_h}{\partial \gamma} \leq 0$. However if $\bar{\alpha}\gamma \geq \sqrt{2}v$, then the firm producing the hardware benefits from piracy. The reason is that this firm indeed loses twice: Once on the licensing fee (see Equation 16), and second because the lower bound for buying the hardware α_1 is getting slightly larger (see Equation 18). But when piracy is large to begin with (as required by condition 22), the change in this lower bound is small. On the other hand, the hardware firm gains considerably by charging more for the hardware at a rate of $\frac{\bar{\alpha}}{2}$ (see Equation 15), and so if $\bar{\alpha}$ is large enough (as required by condition 22), then the overall effect is to increase its profits when piracy increases.

As in the previous case, as a sanity check, we can compute what will be the profits of the hardware producer if it decides not to buy any software from the software provider, and compare it to the profits given in Equation (20). We first note that the software provider develops software if $\pi_s^O \geq 0$. For a range of (γ, C_s) such that $\pi_s^O > 0$ (inner solutions), it is easy to check that $\pi_h^O > \bar{\alpha}v$ for any γ within the range. This is because the hardware firm earns positive licensing fee profits (i.e., $\frac{\bar{\alpha}^2(1-\gamma)}{8} > 0$), and the profits from hardware sales are at least as large as $\bar{\alpha}v$.

Now consider a range of (γ, C_s) such that $\pi_s^O \leq 0$ (corner solutions). Under this condition, the hardware firm has two options: (i) lower the licensing fee and guarantee that $\pi_s^O = 0$, or (ii) abandon software and only sell hardware (and earn $\pi_h = \bar{\alpha}v$). Recall Equation (13). The constraint $\pi_s = 0$ implies that

$$\begin{aligned} \pi_s(f) = 0 &\Leftrightarrow \frac{(\bar{\alpha}(1-\gamma) - f)^2}{4(1-\gamma)} - \frac{C_s}{2} = 0 \\ &\Leftrightarrow f = \bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}, \end{aligned}$$

where the last identity comes from the constraint that the demand for software is nonnegative. Under this f, the demand for software is $\sqrt{\frac{C_s}{2(1-\gamma)}}$. Substituting this licensing fee and Equation (15) into Equation (14) yields

$$\pi_h^O = \begin{cases} \bar{\alpha}v + \left(\bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}\right)\sqrt{\frac{C_s}{2(1-\gamma)}} & \text{if } \gamma < \frac{v}{\bar{\alpha}}, \\ \frac{(\bar{\alpha}\gamma+v)^2}{4\gamma} + \left(\bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}\right)\sqrt{\frac{C_s}{2(1-\gamma)}} & \text{if } \gamma \geq \frac{v}{\bar{\alpha}}. \end{cases} \tag{23}$$

When $\gamma < \frac{v}{\bar{\alpha}}$, we have seen that the hardware firm has an incentive to make the software provider develop software as long as the license profits are nonnegative, which is equivalent to a nonnegative licensing fee. When $\gamma \geq \frac{v}{\bar{\alpha}}$, the optimal f could be negative and the profit loss due to the negative f could be fully compensated by an increase in the profits from hardware (as it increases with γ). However, for a very large γ, the "subsidy" to the software firm through a negative licensing fee becomes too large to be compensated by hardware profits. As a result, the hardware firm will prefer not having software.

The following Proposition summarizes the results of the Full Outsourcing case.

Proposition 2.3. *[Full Outsourcing] For small C_s and γ, the hardware firm will set a positive license fee and the software firm produces software and earn positive profits. As γ increases, the hardware firm's profits could increase in γ. For a large γ, the software producer earns zero profits and the license fee becomes negative. As γ further increases, the hardware firm prefers not to subsidize the software firm, i.e., it only sells hardware.*

We provide complete analysis in Appendix A.2.

2.4 Piracy and endogenous outsourcing decision

Built on the previous two baseline analyses, we now allow the hardware firm to control how much of the software they produce in-house (δ) and how much they outsource ($1 - \delta$). The hardware firm sets the price of the hardware (p_h), the proportion of software developed in house (δ), and the license fee (f). The software firm sets the price of the software (p_s).

From the analyses in the two baseline cases, we know that when the cost of developing software is large or when the degree of piracy is too high, the hardware firm prefers to sell only hardware. Moreover, when γ is too small, all consumers purchase hardware and hardware profits are independent of the degree of piracy. These intuitions still hold in the current scenario, and thus we restrict our attention to the interior solution in the current scenario:

Assumption 2.4. *The cost of developing software is sufficiently low:*

$$C_s \leq C_h \leq \frac{27\bar{\alpha}(\bar{\alpha} - v)}{128}.$$

Assumption 2.5. *The degree of piracy is intermediate: let $\underline{\gamma} \equiv 1 - \frac{128C_h}{27\bar{\alpha}^2}$ and $\overline{\gamma} \equiv 1 - \frac{4(4C_h + C_s)C_hC_s}{(2C_h + C_s)^2\bar{\alpha}^2}$. Then, we assume*

$$\gamma \in \Gamma \equiv \left[\underline{\gamma}, \overline{\gamma}\right].$$

It is easy to check that $C_s \leq C_h$ implies that Γ is non-empty.

We start with the software firm's problem. As we will see below, the software firm's equilibrium software pricing will be identical to the Full Outsourcing case. Given (δ, f), the software firm's problem is

$$\max_{p_s} \, (1-\delta)(p_s - f)\left(\bar{\alpha} - \frac{p_s}{1-\gamma}\right) - \frac{C_s}{2}(1-\delta)^2.$$

The first-order condition with respect to p_s is the same as before, and gives

$$p_s(f) = \frac{\bar{\alpha}(1-\gamma)+f}{2}, \qquad Q_s(f) = \frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)}.$$

Plugging these into π_s, we get

$$\pi_s(\delta, f) \;=\; (1-\delta)\frac{(\bar{\alpha}(1-\gamma)-f)^2}{4(1-\gamma)} - \frac{1}{2}C_s(1-\delta)^2.$$

The profits for the hardware firm consists of three elements: hardware profits (π_h^{hw}), profits from license fees for outsourced software ($\pi_h^{sw,out}$), and in-house software profits ($\pi_h^{sw,in}$). The profit function of the hardware firm is

$$\begin{aligned}\pi_h &= \pi_h^{hw} + \pi_h^{sw,out} + \pi_h^{sw,in} \\ &= \underbrace{p_h\left(\bar{\alpha} - \frac{p_h - v}{\gamma}\right)}_{\text{hardware profits}} + \underbrace{(1-\delta)f\left(\frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)}\right)}_{\text{profits from license fees}} + \underbrace{\delta\frac{\bar{\alpha}(1-\gamma)+f}{2}\left(\frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)}\right) - \frac{1}{2}C_h\delta^2}_{\text{profits from in-house software}}.\end{aligned}$$

The hardware firm maximizes the profits by choosing the price of hardware (p_h), license fee (f), and proportion of in-house software (δ). First, consider the hardware price. The first-order condition with respect to p_h does not involve other endogenous variables (f and δ), and the optimal p_h can be obtained as

$$p_h = \frac{v + \gamma\bar{\alpha}}{2}.$$

The first-order conditions for f and δ are

$$\begin{aligned}\frac{\partial \pi_h}{\partial f} &= \underbrace{-\frac{1}{2(1-\gamma)}\left[(1-\delta)f + \delta\frac{\bar{\alpha}(1-\gamma)+f}{2}\right]}_{\text{decrease in } \pi_h \text{ due to a decrease in software sales}} + \underbrace{\left(\frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)}\right)\left[1-\frac{\delta}{2}\right]}_{\text{increase in } \pi_h \text{ due to an increase in ``price'' } (f \text{ and } p_s)} = 0 \\ \frac{\partial \pi_h}{\partial \delta} &= \underbrace{\left(\frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)}\right)\left[-f + \frac{\bar{\alpha}(1-\gamma)+f}{2}\right]}_{\text{increase in } \pi_h \text{ due to an increase in ``price'' (i.e., } f < p_s)} - \underbrace{C_h\delta}_{\text{decrease in } \pi_h \text{ due to an increase in development cost}} = 0.\end{aligned}$$

First, the first-order condition with respect to f gives

$$f = \frac{\bar{\alpha}(1-\gamma)(1-\delta)}{2-\delta}. \tag{24}$$

Equation (24) implies that the optimal license fee is positive, and this is because we focus on the interior solution. Also, note that for any δ, f is uniquely determined. To see this, notice that

$$\frac{\partial f}{\partial \delta} = -\frac{\bar{\alpha}(1-\gamma)}{(2-\delta)^2} < 0.$$

The above inequality also implies that as the proportion of in-house software increases, the license fee goes down. To see this intuitively, note that the marginal return from f for outsourced software profit ($\frac{\partial \pi_h^{sw,out}}{\partial f}$) and that for in-house software profit ($\frac{\partial \pi_h^{sw,in}}{\partial f}$) are

$$\frac{\partial \pi_h^{sw,out}}{\partial f} = (1-\delta)\frac{\bar{\alpha}(1-\gamma)-2f}{2(1-\gamma)}, \quad \text{and} \quad \frac{\partial \pi_h^{sw,in}}{\partial f} = -\delta\frac{f}{2(1-\gamma)}.$$

The first-order condition with respect to f requires $\frac{\partial \pi_h}{\partial f} = \frac{\partial \pi_h^{sw,out}}{\partial f} + \frac{\partial \pi_h^{sw,in}}{\partial f} = 0$. Also, as Equation (24) implies that the optimal f is strictly positive for $\delta > 0$, the marginal return from f for in-house software is strictly negative (i.e., $\frac{\partial \pi_h^{sw,in}}{\partial f} < 0$). These two observations suggest that at the optimal f, the marginal return from f for outsourced software is strictly positive (i.e, $\frac{\partial \pi_h^{sw,out}}{\partial f} > 0$). Thus, if the hardware chooses to increase the proportion of in-house software, then it should lower the license fee so that it increases the return from additional in-house software.

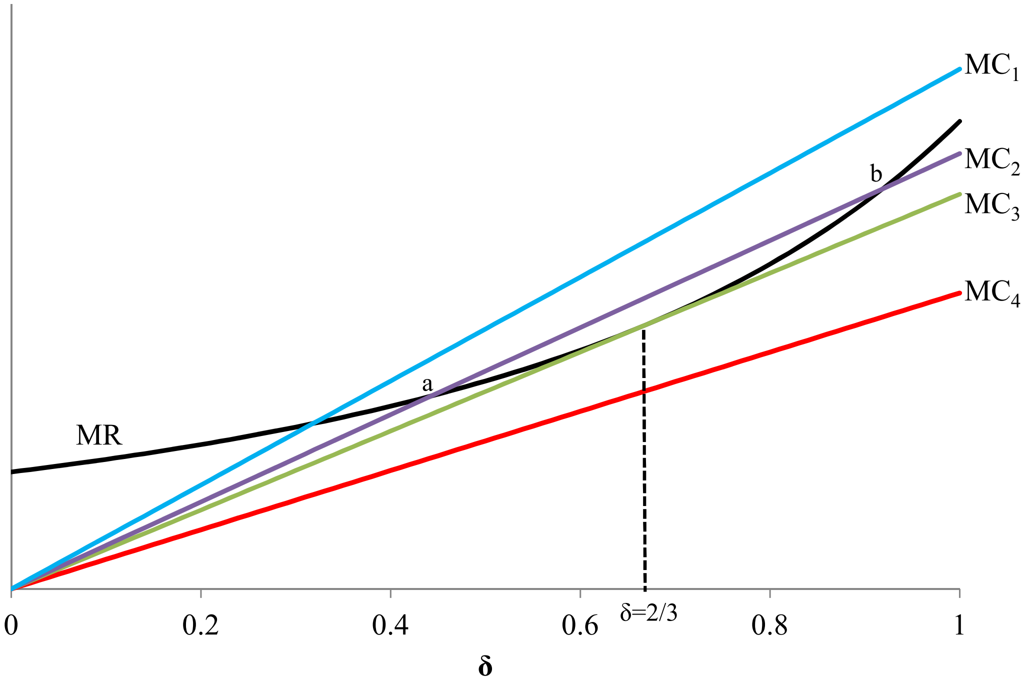
Now, substituting the expression for the optimal f (Equation 24) into the first-order condition for δ, we get

$$\underbrace{\frac{\bar{\alpha}^2(1-\gamma)}{4(2-\delta)^2}}_{\text{MR of in-house software given optimal } f} = \underbrace{C_h\delta}_{\text{MC of in-house software}}. \tag{25}$$

This condition characterizes the optimal δ. The analytical solution is complicated and multiple solutions could exit. However, we can get intuitions implicitly. First, it can be verified that the marginal revenue in (25) is a convex function of δ on [0,1], and the marginal cost is linear in δ. Figure 2 shows a graphical representation of condition (25). We plot the marginal revenue (the convex curve labeled as MR), and four lines for the marginal cost with varying C_h (MC$_1$ to MC$_4$) over δ. The solution(s) of Equation (25) depend on the values of $(\bar{\alpha}, C_h, \gamma)$:

$$\#\text{ solutions} = \begin{cases} 1\ (\delta < \frac{2}{3}) & \text{if } \frac{\bar{\alpha}^2(1-\gamma)}{4C_h} \in (0,1)\ (\text{MC}_1), \\ 2\ (\text{one solution} < \frac{2}{3} \text{ and another} > \frac{2}{3}) & \text{if } \frac{\bar{\alpha}^2(1-\gamma)}{4C_h} \in [1, \frac{32}{27})\ (\text{MC}_2), \\ 1\ (\delta = \frac{2}{3}) & \text{if } \frac{\bar{\alpha}^2(1-\gamma)}{4C_h} = \frac{32}{27}\ (\text{MC}_3), \\ 0 & \text{if } \frac{\bar{\alpha}^2(1-\gamma)}{4C_h} > \frac{32}{27}\ (\text{MC}_4). \end{cases}$$

Assumption 2.5 guarantees that $\frac{\bar{\alpha}^2(1-\gamma)}{4C_h} \leq \frac{32}{27}$, thus at least one solution exists. For MC$_2$, we have two solutions (points a and b in Figure 2). However, we can show that solution b ($\delta > \frac{2}{3}$) is a



Notes: MR in the figure corresponds to the LHS of Equation (25), and MC$_1$-MC$_4$ correspond to the RHS of Equation (25) with varying C_h.

Figure 2: A graphical representation of optimal δ

saddle point, and thus the optimal proportion of in-house software (δ) is less than or equal to $\frac{2}{3}$. We provide more detail on this in Appendix A.3.

Finally, note that the marginal revenue in Equation (25) is decreasing in γ. In Figure 2, a downward shift of the marginal revenue curve moves the solution of δ to the left (for $\delta < \frac{2}{3}$). Thus, we have $\frac{\partial\delta}{\partial\gamma} < 0$ for $\delta < \frac{2}{3}$. This can also be shown by multiplying both sides of Equation (25) by $(2-\delta)^2$ and differentiating with respect to γ:

$$
\begin{aligned}
-\frac{\bar{\alpha}^2}{4} &= \frac{\partial\delta}{\partial\gamma}C_h(2-\delta)^2 + 2\delta C_h(2-\delta)\left(-\frac{\partial\delta}{\partial\gamma}\right) \\
\Leftrightarrow \quad \frac{\partial\delta}{\partial\gamma} &= -\frac{\bar{\alpha}^2}{4C_h(2-\delta)(2-3\delta)}, \qquad \delta \neq \frac{2}{3}.
\end{aligned}
$$

The following Proposition summarizes the main results of the endogenous outsourcing scenario, and we provide details in Appendix A.3.

Proposition 2.6. *Suppose Assumptions 2.4 and 2.5 hold. Then, for $\gamma \in \Gamma$ defined in Assumption*

2.5, there exists a unique optimal strategy (p_h^*, f^*, δ^*) *by the hardware firm that satisfies*

$$p_h^* = \frac{v + \gamma\bar{\alpha}}{2}, \qquad f^* = \frac{\bar{\alpha}(1-\gamma)(1-\delta^*)}{2-\delta^*},$$
$$\delta^*(2-\delta^*)^2 = \frac{\bar{\alpha}^2(1-\gamma)}{4C_h}, \qquad \delta^* \in \left[\frac{C_s}{2C_h + C_s}, \frac{2}{3}\right].$$

Moreover, under this optimal strategy, we have

$$\frac{\partial \delta^*}{\partial \gamma} < 0.$$

That is, as the degree of piracy increases, the hardware firm will use more outsourcing for the production of software.

When γ becomes larger than the upper bound of Γ, the software firm's profits become zero. As before, the hardware firm may be willing to "subsidize" the software firm by lowering the licensing fee, but it will eventually prefer not to have software when γ becomes too large. When γ is lower than the lower bound of Γ, the hardware firm will sell hardware to all consumers. Under this situation, hardware profits are no long a function of γ.

Next, we state that when C_h is low, the hardware firm's profit π_h can increase in piracy when the degree of piracy γ is high. For this to happen for $\gamma \in \Gamma$, we require the following assumption:

Assumption 2.7. *Let* $C_s = \mu C_h$ *for* $\mu \in (0, 1]$.

$$C_h < \frac{(2+\mu)^2(2\bar{\alpha}^2 - (4+\mu)v^2)}{8(4+\mu)\mu}.$$

Proposition 2.8. *Suppose Assumptions 2.4 and 2.5 hold. Then, for* $\gamma \in \Gamma$ *defined in Assumption 2.5 and under Assumption 2.7, there exists a unique* $\hat{\gamma} < \overline{\gamma}$ *such that for* $\gamma > \hat{\gamma}$, $\frac{\partial \pi_h}{\partial \gamma} > 0$. *Furthermore,* $\hat{\gamma} > \frac{\sqrt{2}v}{\bar{\alpha}}$, *i.e., the threshold value of* γ *in the current case is strictly greater than that in the Full Outsourcing case.*

More details are provided in Appendix A.4.

3 Empirical Investigation

Our theoretical examination shows that the proportion of in-house software decreases with the degree of piracy. In this section, we provide some empirical evidence for the external validity of our theoretical prediction using data from the U.S. handheld video game market between 2004 and 2012. Video game markets are a canonical example of two-sided markets in which software firms interact with consumers through platforms (video game consoles/handhelds) (Clements and Ohashi 2005, Dubé, Hitsch, and Chintagunta 2010, Chao and Derdenger 2013, Derdenger and Kumar 2013, Lee 2013, Derdenger 2014). During the sample period, the handheld market was dominated by two major platforms: Nintendo DS (NDS), released in November 2004 by Nintendo, and Sony PlayStation Portable (PSP), released in March 2005 by Sony Computer Entertainment.[4] These two platforms provide a novel empirical setting for investigating our theoretical prediction. First, software titles on NDS and PSP are developed by both the hardware firm (Nintendo/Sony) and third-party software firms (e.g., Activision Blizzard, Electronic Arts, Square Enix).[5] Thus, we can examine the extent to which software titles are developed in-house versus outsourced. Second, these platforms are known to have suffered from software piracy significantly (Fukugawa 2011). According to a study conducted by Computer Entertainment Suppliers Association in Japan in 2010, the estimated total revenues lost due to software piracy on NDS and PSP is $41.7 billion from 2004 to 2009 worldwide.[6] The significant effect of piracy was mainly because of the devices that easily make illegally downloaded software playable on NDS and PSP. For NDS, a small device called the Revolution for DS (R4) made hacking possible. It is a cartridge that can be inserted into NDS and allow downloaded ROMs to be booted on NDS from a microSD card. For Sony PSP, hacking was made possible via a Pandora battery and a Magic Memory Stick. Eventually, Nintendo took a legal action to stop the sales of R4, but Sony did not.

3.1 Data

This section describes the empirical measures used for our empirical examination. Our goal is to examine the effect of software piracy on the proportion of in-house software. We collected data on measures for (1) the proportion of in-house software (dependent variable), (2) the degree of software piracy (key independent variable), and (3) control variables. The unit of our analysis is (platform, month), and our sample size is 172.[7] In what follows, we will explain each set of measures.

[4] We note that our theoretical model examines a monopoly platform's decision. Although the empirical application in this section has two platforms, NDS and PSP are highly differentiated from one another. PSP's target consumer segment was conventional gamers who appreciate high-quality graphics in a portable device, and NDS went after children and casual gamers and offered a new way of playing games with touch screen and pen.

[5] We note that neither Nintendo nor Sony developed software for its rival's platform.

[6] https://www.engadget.com/2010/06/06/cesa-calculates-gaming-industry-losses-due-to-portable-piracy/.

[7] Our theoretical prediction is based on a static model, but our empirical measures are observed at the monthly level. Although developing a dynamic model is beyond the scope of this paper, we conjecture that our prediction on the outsourcing decision will extend to a dynamic setting. In a dynamic variant of our model, consumers in subsequent periods will have a lower α, which reduces the equilibrium hardware price over time (Nair 2007, Liu 2010). If consumers are forward-looking, they might delay purchase and this will increase α_1 in Figure 1 in period 1. However, as we saw, since the optimal δ is independent of p_h, we expect that the impact of γ on δ will still remain negative.

3.1.1 Measures for the dependent variable

To measure the extent to which software is developed in-house versus outsourced, we obtain data from NPD on monthly sales of all software titles released on NDS and PSP from their inception to February 2012. For each software, we use its publisher identity for grouping software into in-house (when the publisher is the platform provider) and outsourced (when the publisher is a third-party software provider).[8] For NDS, 1,777 software titles are released during the sample period, and 109 titles (6.1%) are by Nintendo (i.e., in-house). Examples of top selling in-house software titles include *New Super Mario Bros.*, *Mario Kart DS*, and *Pokémon Diamond Version*, and examples of top selling outsourced software include *Guitar Hero on Tour Bundle* (by Activision Blizzard), *Lego Star Wars: The Complete Saga* (by LucasArts), and *Cooking Mama* (by Majesco Entertainment). For PSP, we observe 626 titles and 77 of them (12.3%) are by Sony. Examples of in-house software include *God of War: Chains of Olympus*, *SOCOM U.S. Navy SEALs: Fireteam Bravo*, and *Ratchet & Clank: Size Matters*, and examples of outsourced software include *Grand Theft Auto: Liberty City Stories* (by Take-Two Interactive), *Need for Speed: Most Wanted* (by Electronic Arts), and *Star Wars: Battlefront II* (by LucasArts).

Table 1 presents summary statistics on monthly software-related measurements for NDS and PSP. The number of observations for NDS is 88 months, and that for PSP is 84 months. Our main dependent variable is the proportion of in-house software. We create two measures using (1) the number of "active" software, and (2) software revenues. "Active" software in a given month is defined as software which has positive sales in that month. Video game software sales typically peaks in the release month, declines quickly over time, and eventually becomes zero. We count the number of software with positive sales in each month, for both in-house and outsourced. The first panel of Table 1 shows that the average number of active software in any given month is 662.2 titles for NDS and 317.7 titles for PSP. The average number of active in-house software is 51 titles (11.8%) for NDS and 44.7 titles (16.8%) for PSP. These numbers are slightly higher than the numbers described earlier (6.1% for NDS and 12.3% for PSP). This is mainly because in-house video games tend to stay in the market longer than outsourced video games. We use these proportions as one of the measures for the proportion of in-house software development.

Another measure is derived from the proportion of in-house software revenues. Our theoreti-

[8] We note that it is possible that in-house software is developed by an independent software developer and published by a platform provider (see Gil and Warzynski 2015, Ishihara and Rietveld 2017). In this study, we focus on publisher identity because the decision to release a game is made by publishers.

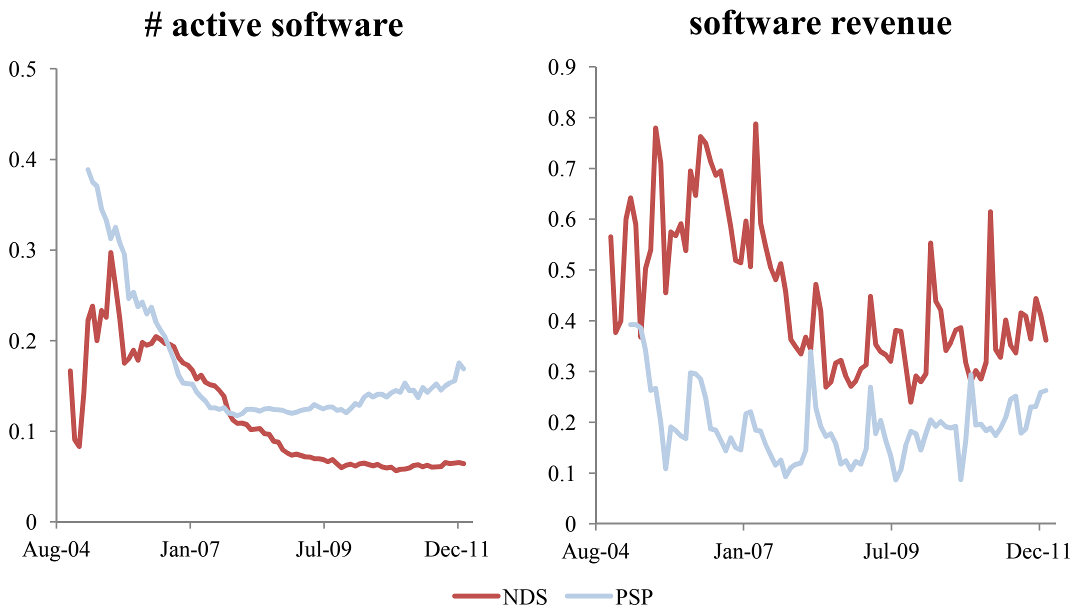
Platform	Variable	Average	SD	Min	Max
# active software					
NDS	all software	662.2	489.7	6	1418
(N=88)	in-house software	51.4	25.9	1	92
	in-house proportion	11.8%	6.19%	5.65%	29.7%
PSP	all software	317.7	148.9	18	490
(N=84)	in-house software	44.7	17.5	7	66
	in-house proportion	16.8%	6.88%	11.7%	38.9%
software revenue					
NDS	all software	\$90.2m	\$88.5m	\$6.76m	\$470.1m
(N=88)	in-house software	\$35.9m	\$31.2m	\$2.48m	\$148.3m
	in-house proportion	44.8%	14.3%	23.9%	78.8%
PSP	all software	\$29.8m	\$26.8m	\$2.86m	\$147.2m
(N=84)	in-house software	\$5.53m	\$4.97m	\$0.64m	\$27.1m
	in-house proportion	19.2%	6.73%	8.65%	39.3%

Notes: The number of observations (N) indicates the number of months we observe data for NDS (88 months) and PSP (84 months).

Table 1: Summary statistics on monthly software-related measures

cal model does not account for software differentiation between in-house software and outsourced software and also that within each group of software. However, in reality, software titles are differentiated in quality and other observed attributes. The proportion of in-house software revenues account for the average difference between the two groups of software, and can be thought of as a weighted average version of the first dependent variable measure. The second panel of Table 1 shows the summary statistics on software revenues. The average monthly software revenue (all active titles) is USD 90.2 million for NDS and 29.8 million for PSP. The average monthly revenue for in-house software is USD 35.9 million (44.8%) for NDS and USD 5.5 million (19.2%) for PSP. We note that the proportion of in-house software revenues for NDS is significantly higher than that for PSP, despite the fact that the proportion of the number of active in-house software for NDS is lower than that for PSP. This is because Nintendo owns several hugely successful intellectual properties such as Mario franchise and Pokémon franchise which disproportionately generate huge revenues as compared to outsourced video games.

The two panels in Figure 3 plot our dependent measures over time. The left panel shows the monthly proportion of in-house software based on the number of active software, for NDS and PSP. For NDS, it fluctuates between 0.1 and 0.3 right after the platform release, gradually declines, and reaches a steady level of around 0.07. For PSP, it starts high (around 0.4) and quickly declines until around April 2007, and slightly increases from there. Common in both platforms is



Notes: The data period is from November 2004 to February 2012 for NDS, and from March 2005 to February 2012 for PSP.

Figure 3: Monthly proportion of in-house software over time

that the proportion tends to be higher in the earlier platform lifecycle than in the later platform lifecycle. This might be because the platform providers needed to boost the sales of handheld devices (platform adoption) in the early platform lifecycle, so as to enjoy the positive indirect network feedback loop in the subsequent periods (e.g., increases software titles, which in turn increases hardware sales). To do so, instead of relying on third-party software firms to release video game titles in the early product lifecycle, the platform providers release their own in-house games. Since our theoretical model is static and does not account for such a dynamic incentive shift, we will control for the effect of platform lifecycle by including the cumulative number of hardware sales as a control variable (we will discuss below).

The right panel shows the monthly proportion of in-house software revenues for NDS and PSP. This measure fluctuates more significantly than the measure based on the number of active software. Spikes are typically associated with a release of blockbuster in-house software. There is also a mild seasonal effect: the in-house software proportion tends to be high in March, April, and May than in other months for both NDS and PSP. For NDS, the proportion is also relatively high in August and September. We will include monthly dummies as controls to account for such variation. Finally, we note that for NDS, the proportion is on average higher in the earlier platform lifecycle than in the later platform lifecycle, consistent with the trend of the other measure on the left panel. However, for PSP, the proportion fluctuates around 0.2 throughout the platform lifecycle, despite the fact that the proportion on the left panel shows a declining pattern over time. This is mainly because blockbuster software in PSP tends to be released by third-party firms (e.g., Take-Two Interactive and Electronic Arts), and many of those titles were multi-platform releases such as PlayStation 2 (e.g., Grand Theft Auto: Liberty City Stories). Since the extra cost of porting games from PlayStation 2 to PSP is relatively low, third-party firms were able to release blockbuster titles in the early lifecycle of PSP. However, NDS was an innovative handheld device with touch screen and pen, and many NDS games were designed to take advantage of this unique feature of NDS. As a result, third-party firms did not enjoy the multi-platform strategy that involves NDS as much as they did for PSP.

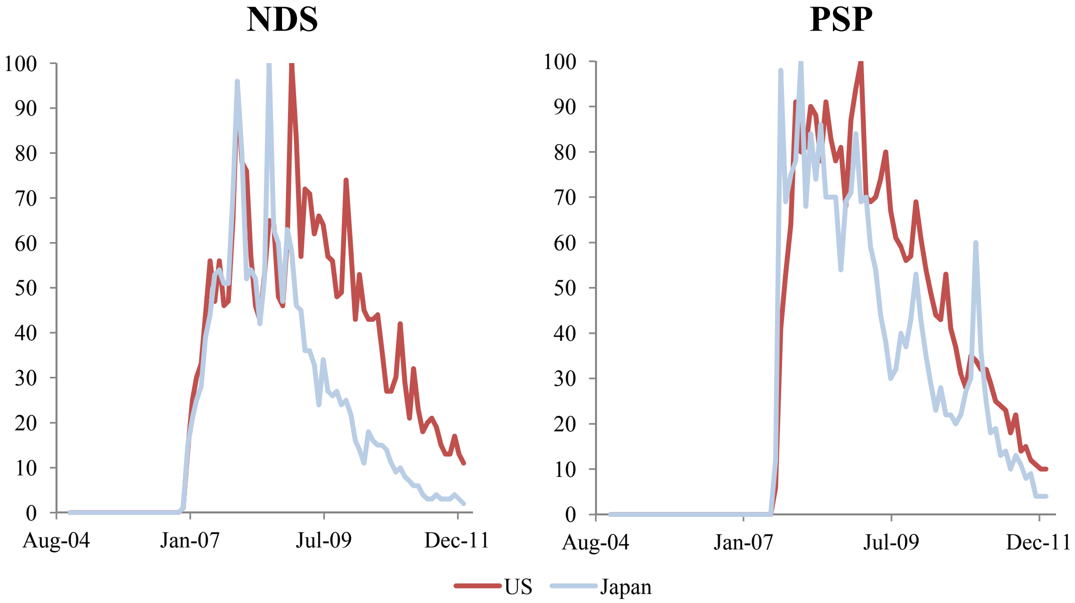
3.1.2 Measures for the key independent variable

The key independent variable of our regression analysis is the degree of piracy. In the theoretical model, we operationalize it as a lowered quality of software relative to a legal version (γ). The lowered quality may be due to a variety of reasons, but one such factor is the cost of using pirated software. For NDS and PSP, in order to play illegally downloaded games, consumers need to know how to use the device (R4 for NDS and Pandora battery for PSP) for hacking the system of the handheld devices. Since this is an illegal act, most consumers search information online and find out how-to. Such information is initially spread and shared only among hardcore hackers. But what was unique about NDS and PSP software piracy is that the device made hacking so popular and accessible to regular gamers that the information on the device became widely spread online.[9] As more websites appear and explain how to use the device, the cost of using pirated software decreases. In other words, we could use the volume/accessibility of online information as a proxy for the (inverse) cost of using pirated software.

In our regression analysis, we use Google Trends' (relative) search volume to approximate the accessibility of online information on how to use the device.[10] Since Google Trends is a result of consumers' interest in a certain keyword, it may not exactly match with the information accessibility. Thus, our key identification assumption is that as more information becomes available, the chance of finding information becomes higher. As a result, the search volume increases. Under this

[9] In fact, using R4 for NDS became widely well-known even among primary school children in Japan, and many parents (who do not play video games) did not realize it is illegal and they made inquiries at video game shops as to how to use the device to make downloaded games playable on NDS.

[10] Google Trends (`https://trends.google.com/trends/`) shows how often a particular keyword is searched relative to the total search volume.



Notes: For NDS, the search volume for "ds r4" is shown monthly from November 2004 to February 2012. For PSP, the search volume for "psp pandora battery" is monthly shown from March 2005 to February 2012. The search volume for Japan uses the Japanese equivalent of these combined keywords.

Figure 4: Google Trends monthly search volume over time

assumption, consumers' interest in the device (as captured by Google Trends) will capture variation in the accessibility of information on how to use the device.

Specifically, we use the term "ds r4" and "psp pandora battery" as the search keywords for NDS and PSP, respectively.[11] We made the inquiry separately and obtain the monthly search volume over our sample period using the U.S. as the specified region. The value of the search volume is normalized in a way that the value, 100, is assigned to the peak search volume during the entire period. Since we retrieved data separately for NDS and PSP, both series will have 100 as the peak search volume value. Thus the comparison of the search volume between NDS and PSP is meaningless, and only the time-series variation within each series matters. In our regression, we include the platform fixed-effect as a control for adjusting the level effect.

Figure 4 shows the search volume for "ds r4" (left panel) and "psp pandora battery" (right panel) over the sample period. In addition to the search volume in the U.S., we plot the search volume restricted to Japan (in Japanese equivalent of the keywords). As we will explain below, the search volume obtained from Japan will be used as an instrument in our regression. The search

[11] We did not use "nds" because "ds" was a more widely used term for referring to Nintendo DS. For PSP, we did not include a term "Magic Memory Stick," mainly because it makes the search volume significantly smaller. Also, consumers only need to buy a Pandora battery (Magic Memory Stick can be easily made by downloading software and storing it in a regular Memory Stick).

volume for NDS is essentially zero for about two years after NDS release (until November 2006), and increases sharply during 2007. In this year, NDS software piracy became a serious issue for software firms and some software firms started embedding a code in software that prevents pirates from playing an illegally downloaded version.[12] However, such a prevention code was often cracked by hackers a few days after release, and it never became a real solution. In 2008, Nintendo started taking a legal action worldwide against retailers who sell R4.[13] For example, Tokyo District Court ruled against the distribution of R4 in February 2009. This coincides with a significant decline in the search volume in Japan. In the U.S., the legal action was not taken during our sample period, which is consistent with the longer lasting U.S. search volume.

We observe similar patterns for PSP. The search volume is zero until July 2007, and sharply increases in the rest of 2007. The trend happened in Japan slightly earlier than in the U.S. Sony, instead of taking a legal action, constantly introduced system software updates as well as new hardware models that embed better protection against piracy.

In summary, we use the variation in search volume as a proxy for the degree of piracy. Based on our theoretical prediction, we expect that a higher search volume leads to a lower proportion of in-house software.

3.1.3 Control variables

In order to control for other factors that might influence the proportion of in-house software, we collect additional data and generate control variables. First, we obtain data from NPD on monthly hardware unit sales for NDS and PSP from their inception to February 2012. We then compute the cumulative number of hardware unit sales and include this variable to control for the effect of platform lifecycle on the dependent measure. We plot the cumulative hardware sales over time in the left panel of Figure 5. Overall, the difference between NPD and PSP was small in the beginning, but gets wider as time goes one. A periodic jump is due to the Christmas seasonal effect. As we discussed above, platform's lifecycle plays an important role in influencing the proportion of in-house versus outsourced software. At the beginning of the lifecycle, third-party software firms may be skeptical about the success of a platform. Moreover, the cost of developing software for a new platform may be high because programmers may not be familiar with the development environment for making software for the new platform. As time goes on, if the platform turns out to be a success (which is captured by a large number of cumulative hardware sales), software firms will have more

[12] See, e.g., https://www.engadget.com/2008/03/11/square-enix-thanks-pirates-for-playing-ffcc/.

[13] https://ap.nintendo.com/news/press.jsp.

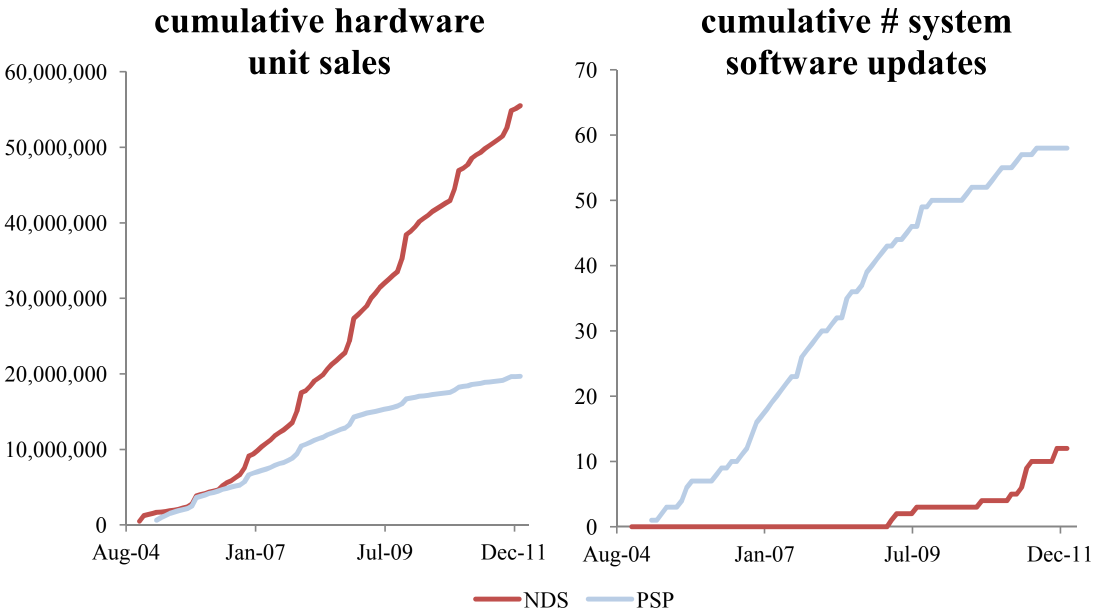


Figure 5: Cumulative hardware unit sales and cumulative number of system software updates over time

incentive to release software on the platform.

Second, we obtain monthly occurrence data on system software update releases for both NDS and PSP. System software is similar to an operating system in computer, and controls all functionalities available on a handheld device. System software updates are not necessarily targeted against software piracy (e.g., fixing known bugs, adding new features to the device), but can also be used for embedding piracy protection features. The right panel of Figure 5 shows the cumulative number of system software updates over time for NDS and PSP. As we mentioned earlier, Sony was active in providing updates for improving piracy protection but Nintendo was not. If software firms expect new system software updates by a platform in a given month, and if the updates are related to piracy protection, they may align the introduction of a software title with the system software updates. Moreover, in general, they may find it more attractive to introduce software to a platform with more system software updates. Thus, we include both monthly and cumulative system software updates as control variables.

Finally, as we discussed above, we include monthly dummies to control for popular months for introducing software. Since we have only two platforms, we are not able to control for calendar time fixed effects. We tried specifications with year fixed effects but found that they are highly correlated with the (logged) cumulative hardware sales and created a multicollinearity issue. Thus, we dropped year fixed effects.

3.2 Results

Our econometric model is

$$y_{it} = \alpha G_{it} + \beta X_{it} + \mu_i + \epsilon_{it},$$

where the subscripts i and t index platform and time, y_{it} is a measure for the proportion of in-house software (either based on the number of active software or software revenue), G_{it} is Google Trends' search volume for hacking devices (a proxy for the degree of piracy), X_{it} is a vector of observed controls (logged cumulative hardware sales, system software updates, and monthly dummies), μ_i is a platform fixed effect, and ϵ_{it} is an error term. Our main parameter of interest is α, the effect of piracy on the proportion of in-house software. Our theoretical result predicts that α is negative.

The key econometric issue in estimating the above model is that G_{it} and ϵ_{it} may be correlated. For example, it is possible that Google Trends U.S. search volume may be correlated with a new release of blockbuster in-house games in the U.S. That is, when a popular game is released, consumers might search for information about hacking device so that they can play it for free illegally, which makes the search volume endogenous. We thus estimate the model using the Two-Stage Least Squares. We use Google Trends Japan's search volume for hacking devices (in Japanese) as an instrument for the U.S. measure. As we saw in Figure 4, these two measures are highly correlated. Also, as a set of video games available and released in Japan in a given month are different from those in the U.S., we expect that Google Trends Japan's search volume is uncorrelated with the error term.[14]

We report the parameter estimates of the model in Table 2. The first panel of Table 2 shows a set of results based on the number of active software and the second panel is based on software revenues. In each panel, we estimated four models that differ in the set of control variables. For all the specifications, we compute the standard errors based on the heteroskedasticity- and autocorrelation-consistent (HAC) variance estimates (Newey and West 1987) with the Bartlett kernel and the bandwidth of two. Also, in order to check the validity of the instruments, we report the first-stage F-statistic of excluded instruments. The F-statistics suggest that our instruments are not weak (Staiger and Stock 1997).

We first discuss the results in the first panel (the number of active software). Model 1 includes

[14] Some video games are released both in the U.S. and Japan, but not necessarily in the same month. Most games have either U.S. or Japan as a primary target market, and based on the performance in the primary market, they may also be released in the other market. Even popular games that are targeted at both markets from the beginning may not be released in the same month. For example, *Mario Kart DS*, one of the best-selling Nintendo DS games, was released in November 2005 in the U.S., and in December 2005 in Japan.

Table 2: Regression Results

DV = in-house proportion based on # active software

Variable	Model 1	Model 2	Model 3	Model 4
Google trends US (piracy)	-4.61e-4**	-4.42e-4**	-4.55e-4**	-4.30e-4**
	(1.10e-4)	(1.03e-4)	(1.14e-4)	(1.06e-4)
Cumulative hardware sales (logged)	-0.046**	-0.046**	-0.043**	-0.042**
	(0.006)	(0.006)	(0.008)	(0.008)
Software updates		-0.008*		-0.011**
		(0.003)		(0.004)
Cumulative software updates			-4.32e-4	-0.001
			(3.97e-4)	(3.92e-4)
Platform FE	Yes	Yes	Yes	Yes
Month FE	Yes	Yes	Yes	Yes
R-squared	0.783	0.788	0.787	0.795
First-stage F-statistic	73.2	69.5	70.9	67.1
# observations	172	172	172	172

DV = in-house proportion based on software revenue

Variable	Model 1	Model 2	Model 3	Model 4
Google trends US (piracy)	-8.46e-4**	-8.21e-4**	-8.79e-4**	-8.78e-4**
	(2.94e-4)	(2.94e-4)	(2.52e-4)	(2.57e-4)
Cumulative hardware sales (logged)	-0.049**	-0.049**	-0.070**	-0.070**
	(0.009)	(0.009)	(0.014)	(0.014)
Software updates		-0.011		-0.001
		(0.008)		(0.007)
Cumulative software updates			0.003**	0.001
			(0.001)	(0.001)
Platform FE	Yes	Yes	Yes	Yes
Month FE	Yes	Yes	Yes	Yes
R-squared	0.773	0.775	0.804	0.804
First-stage F-statistic	73.2	69.5	70.9	67.1
# observations	172	172	172	172

Notes: Standard errors are reported in parentheses. We compute the standard errors based on the heteroskedasticity- and autocorrelation-consistent (HAC) variance estimates (Newey and West 1987) with the Bartlett kernel and the bandwidth of two. * and ** indicate 5% and 1% significance, respectively.

Google Trends U.S. search volume, logged cumulative hardware sales, platform and month fixed effects. We find that the search volume has a negative and significant effect on the proportion of in-house software, which supports our theoretical prediction that a higher degree of piracy reduces the proportion of in-house software. The (logged) cumulative hardware sales has a negative and significant effect, suggesting that we see more outsourced software in the later platform lifecycle. This result is consistent with our earlier discussion that platform providers might have an incentive to introduce in-house software to boost hardware sales in the early platform lifecycle. Also, software firms may have more incentive to introduce software to a platform with a larger customer base.

In Models 2-4, we add control variables related to system software updates to Model 1. First, Model 2 adds the monthly number of system software updates to Model 1. The effect of the search volume continues to be negative and significant. We find that the monthly number of system software updates has a negative and significant effect on the proportion of in-house software. This result is consistent with our discussion above that software firms might find it profitable to align their software release with the timing of system software updates. Model 3 adds the cumulative number of system software updates to Model 1. Once again, the effect of the search volume has a negative and significant effect. However, we find that the cumulative number of system software updates is insignificant. This may be because the general difference in efforts for piracy protection between the two platforms may be stable over time and captured by the platform fixed effects. Finally, in Model 4, we added both monthly and cumulative system software updates. Our main conclusion remains unchanged.

In the second panel, we uses the proportion of in-house software revenue as the dependent variable. We run the same four specifications. Throughout the four specifications, we find that the search volume has a negative and significant effect on the proportion of in-house software revenue, supporting our theoretical prediction. The (logged) cumulative hardware sales is also negative and significant, suggesting the importance of platform lifecycle in explaining the proportion. However, control variables based on system software updates are mostly insignificant, except for Model 3, where we find that the cumulative number of system software updates has a positive and significant effect on the proportion of in-house software. This is mainly because in the later periods of our sample period, we observe an increasing trend of the proportion of in-house software revenue (right panel of Figure 3), especially for NDS. This trend coincides with the system software updates by NDS (right panel of Figure 5). Interestingly, we do not observe such trend when we use the proportion based on the number of active software (left panel of Figure 3), and this is probably why we did not find a significant effect of the cumulative number of system software updates. The positive effect when the proportion is based on software revenue may because, as a result of increasing piracy protection via system software updates for NDS, consumers became more likely to buy popular in-house software, which pushed the proportion of in-house software revenue.

Overall, throughout our empirical analyses, we consistently find a significant negative effect of piracy on the proportion of in-house software, providing the external validity of our theoretical prediction: the proportion of in-house software decreases as the degree of software piracy increases.

4 Discussion and Conclusion

In this paper we examine the role of software piracy in the outsourcing decision of a platform provider. In particular, we look at a hardware producer such as Sony, that has to make a software outsourcing decision, that is how many games to produce in-house, and how many to outsource from a third-party software provider, where the games are routinely pirated, and that level of piracy has to be taken into account in the outsourcing decision. In such markets, there is a built-in tension between the hardware and software firms with respect to piracy: As the hardware cannot be pirated, the hardware firm indirectly benefits from piracy (up to a level) since all pirates have to purchase the hardware.

Using a vertical differentiation model, as well as empirical study using data from the U.S. handheld video game market, we find that an increase in piracy increases the level of outsourcing of the hardware provider, as well as increasing its profits. The reason for the increase in outsourcing as a response to increase in piracy occurs because as the degree of piracy increases, the loss in in-house software production due to piracy increases. Although the profit margin from in-house software is higher than license fees, the platform provider benefits from shifting software profits from in-house to license fees because the negative marginal impact of piracy can be reduced.

This main result occurs when the level of piracy is intermediate, between a lower and an upper bound. When the level of piracy is smaller than the lower bound, the hardware firm will sell hardware to all consumers, and the hardware profits are no longer a function of piracy. When the level of piracy becomes larger than the upper bound, the software firm's profits vanish, and the hardware firm will be willing to subsidize the software firm by lowering the licensing fee up to a limit of piracy, beyond which the software market disintegrates.

These results point at a major difficulty for the software producers: In practice, much of the anti-piracy measures are within the realm of the hardware producer. These include measures such as developing new models that prevent modification of hardware (Fukugawa 2011).[15] However, in intermediate levels of piracy, as piracy becomes more prevalent, the hardware firms profits increase, and at the same time it shifts the burden to the software firms by outsourcing more. It has no incentive to stop piracy at these levels. Only when the level of piracy becomes acute (as it did in the US in 2008) do the hardware firm's incentives align with the software firm so that actions is taken in the form of change of hardware or legal action.

[15] While software firms tried to embed codes in games that prevented pirates from playing pirated versions, such prevention codes were quickly cracked by hackers and became useless.

References

Chao, Yong, Timothy Derdenger. 2013. Mixed Bunding in Two-Sided Markets in the Presence of Installed Base Effects. *Management Science* **59**(8) 1904–1926.

Church, Jeffrey, Neil Gandal. 1992. Integration, Complementary Products, and Variety. *Journal of Economics & Management Strategy* **1**(4) 651–675.

Clements, Matthew T., Hiroshi Ohashi. 2005. Indirect Network Effects and the Product Cycle: Video Games in the U.S., 1994-2002. *Journal of Industrial Economics* **53**(4) 515–542.

Conner, Kathleen R., Richard P. Rumelt. 1991. Software Piracy: An Analysis of Protection Strategies. *Management Science* **37**(2) 125–139.

Derdenger, Timothy. 2014. Technological Tying and the Intensity of Price Competition: An Empirical Analysis of the Video Game Industry. *Quantitative Marketing and Economics* **12**(2) 127–165.

Derdenger, Timothy, Vineet Kumar. 2013. The Dynamic Effects of Bundling as a Product Strategy. *Marketing Science* **32**(6) 827–859.

Dubé, Jean-Pierre H., Gunter J. Hitsch, Pradeep K. Chintagunta. 2010. Tipping and Concentration in Markets with Indirect Network Effects. *Marketing Science* **29**(2) 216–249.

Fukugawa, Nobuya. 2011. How Serious is Piracy in the Videogame Industry? *Empirical Economics Letters* **10**(3) 225–233.

Gil, Ricard, Frederic Warzynski. 2015. Vertical Integration, Exclusivity, and Game Sales Performance in the US Video Game Industry. *Journal of Law, Economics, and Organization* **31**(1) i143–i168.

Givon, Moshe, Vijay Mahajan, Eitan Muller. 1995. Software Piracy: Estimation of Lost Sales and the Impact on Software Diffusion. *Journal of Marketing* **59**(1) 29–37.

Ishihara, Masakazu, Joost Rietveld. 2017. The Effect of Acquisitions on Product Innovativeness, Quality, and Sales Performance: Evidence from the Console Video Game Industry (2002-2010). Working paper.

Jain, Sanjay. 2008. Digital Piracy: A Competitive Analysis. *Marketing Science* **27**(4) 610–626.

Lahiri, Atanu, Debabrata Dey. 2013. Effects of Piracy on Quality of Information Goods. *Management Science* **59**(1) 245–264.

Lee, Robin S. 2013. Vertical Integration and Exclusivity in Platform and Two-Sided Markets. *American Economic Review* **103**(7) 2960–3000.

Liu, Hongju. 2010. Dynamics of Pricing in the Video Game Console Market: Skimming or Penetration? *Journal of Marketing Research* **47**(3) 428–443.

Nair, Harikesh. 2007. Intertemporal Price Discrimination with Forward-Looking Consumers: An Application to the US Market for Console Video-Games. *Quantitative Marketing and Economics* **5**(3) 239–292.

Newey, Whitney K., Kenneth D. West. 1987. A Simple, Positive Semi-Definite, Heteroscedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica* **55**(3) 703–708.

Peitz, Martin. 2004. A Strategic Approach to Software Protection: Comment. *Journal of Economics & Management Strategy* **13**(2) 371–374.

Rasch, Alexander, Tobias Wenzel. 2013. Piracy in a Two-Sided Software Market. *Journal of Economic Behavior & Organization* **88** 78–89.

Rasch, Alexander, Tobias Wenzel. 2015. The Impact of Piracy on Prominent and Non-prominent Sofwtare Developers. *Telecommunications Policy* **39**(8) 735–744.

Rochet, Jean-Charles, Jean Tirole. 2006. Two-Sided Markets: A Progress Report. *RAND Journal of Economics* **37**(3) 645–667.

Rysman, Marc. 2009. The Economics of Two-Sided Markets. *Journal of Economic Perspectives* **23**(3) 125–143.

Shy, Oz, Jacques-Francǫis Thisse. 1999. A Strategic Approach to Software Protection. *Journal of Economics & Management Strategy* **8**(2) 163–190.

Sinha, Rajiv K., Fernando S. Machado, Collin Sellman. 2010. Don’t Think Twice, It’s All Right: Music Piracy and Pricing in a DRM-Free Environment. *Journal of Marketing* **74**(2) 40–54.

Staiger, Douglas, James H. Stock. 1997. Instrumental Variables Regression with Weak Instruments. *Econometrica* **65**(3) 557–586.

Takeyama, Lisa N. 1994. The Welfare Implications of Unauthorized Reproduction of Intellectual Property in the Presence of Demand Network Externalities. *Journal of Industrial Economics* **42**(2) 155–166.

Vernik, Dinah A., Devavrat Purohit, Preyas S. Desai. 2011. Music Downloads and the Flip Side of Digital Rights Management. *Marketing Science* **30**(6) 1011–1027.

A Appendix

A.1 Proof of Proposition 2.2

The hardware firm's problem is formulated as

$$\max_{p_h,p_s} \; p_h Q_h(p_h) + p_s Q_s(p_s) - \frac{C_h}{2},$$

subject to (1) $Q_h(p_h) \geq Q_s(p_s)$, and (2) $Q_h(p_h) \leq \bar{\alpha}$.[16] The Lagrangian is then given by

$$L(p_h, p_s, \lambda) = p_h Q_h(p_h) + p_h Q_h(p_h) + p_s Q_s(p_s) + \lambda_1\big(Q_h(p_h) - Q_s(p_s)\big) + \lambda_2\big(\bar{\alpha} - Q_h(p_h)\big),$$

and the Kuhn-Tucker conditions are

$$\begin{aligned}
\frac{\partial L}{\partial p_h} &= Q_h(p_h) + p_h Q_h'(p_h) + \lambda_1 Q_h'(p_h) - \lambda_2 Q_h'(p_h) = 0\\
\frac{\partial L}{\partial p_s} &= Q_s(p_s) + p_s Q_s'(p_s) - \lambda_1 Q_s'(p_s) = 0\\
Q_h(p_h) \geq Q_s(p_s) &\quad \lambda_1 \geq 0 \quad \lambda_1\big(Q_h(p_h) - Q_s(p_s)\big) = 0\\
Q_h(p_h) \leq \bar{\alpha} &\quad \lambda_2 \geq 0 \quad \lambda_2\big(\bar{\alpha} - Q_h(p_h)\big) = 0
\end{aligned}$$

We solve this set of inequalities and equations. Below we examine every possible case.

1. When both constraints (1) and (2) are not binding ($\lambda_1 = \lambda_2 = 0$): This is the case with an interior solution. We have

   $$\begin{aligned}
   p_h &= \frac{\bar{\alpha}\gamma + v}{2}, \qquad p_s = \frac{\bar{\alpha}(1-\gamma)}{2}\\
   Q_h &= \frac{\bar{\alpha}\gamma + v}{2\gamma}, \qquad Q_s = \frac{\bar{\alpha}}{4}.
   \end{aligned}$$

   Constraint (1) is satisfied for any γ because $v > 0$. Constraint (2) implies that $\gamma \geq \frac{v}{\bar{\alpha}}$, and Assumption 2.1 (i.e., $\bar{\alpha} > \sqrt{2}v$) implies that the range of γ that supports this scenario is non-empty.

   With the optimal p_h and p_s, the hardware firm's profit is

   $$\pi_h = \frac{(\bar{\alpha}\gamma + v)^2}{4\gamma} + \frac{\bar{\alpha}^2(1-\gamma)}{4} - \frac{C_h}{2}.$$

   Now we check the condition for $\pi_h \geq \bar{\alpha}v$, i.e., the hardware firm prefers to have software. This condition is equivalent to

   $$(\bar{\alpha}^2 - 2\bar{\alpha}v - 2C_h)\gamma + v^2 \geq 0.$$

[16] For simplicity, we drop some of the constraints that will obviously not bind (e.g., $p_h, p_s \geq 0$).

When $C_h \leq \frac{\bar{\alpha}(\bar{\alpha}-2v)}{2}$, $\pi_h > \bar{\alpha}v$ for any γ because $v > 0$. When $C_h > \frac{\bar{\alpha}(\bar{\alpha}-2v)}{2}$, $\pi_h \geq \bar{\alpha}v$ for $\gamma \leq \frac{v^2}{2C_h - \bar{\alpha}(\bar{\alpha}-2v)}$. For the latter case to be non-empty, we need $\frac{v}{\bar{\alpha}} \leq \frac{v^2}{2C_h - \bar{\alpha}(\bar{\alpha}-2v)}$, or $C_h \leq \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$.

In summary, this scenario is supported under the following conditions: (i) $C_h \leq \frac{\bar{\alpha}(\bar{\alpha}-2v)}{2}$ and $\gamma \in \left[\frac{v}{\bar{\alpha}}, 1\right)$; (ii) $C_h \in \left(\frac{\bar{\alpha}(\bar{\alpha}-2v)}{2}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}\right]$ and $\gamma \in \left[\frac{v}{\bar{\alpha}}, \frac{v^2}{2C_h - \bar{\alpha}(\bar{\alpha}-2v)}\right]$.

2. When only constraint (2) is binding ($\lambda_1 = 0$ and $\lambda_2 \geq 0$): This is the case with $Q_s(p_s) < Q_h(p_h) = \bar{\alpha}$ (everyone buys hardware). We have

$$p_h = v, \quad p_s = \frac{\bar{\alpha}(1-\gamma)}{2}$$
$$Q_h = \bar{\alpha}, \quad Q_s = \frac{\bar{\alpha}}{4}.$$

   Constraint (1) is satisfied for any γ. Since constraint (2) is binding, we need $\lambda_1 \geq 0$, which is equivalent to $\gamma \leq \frac{v}{\bar{\alpha}}$. Assumption 2.1 implies that the range of γ that supports this scenario is non-empty.

   With the optimal p_h and p_s, the hardware firm's profit is

$$\pi_h = \bar{\alpha}v + \frac{\bar{\alpha}^2(1-\gamma)}{4} - \frac{C_h}{2},$$

   and it is easy to check that $\pi_h \geq \bar{\alpha}v$ for $\gamma \leq 1 - \frac{2C_h}{\bar{\alpha}^2}$. In order for the range of γ that supports this scenario to be non-empty, we need $1 - \frac{2C_h}{\bar{\alpha}^2} > 0$, or $C_h < \frac{\bar{\alpha}^2}{2}$.

   In summary, the following conditions support this scenario: (i) $C_h \leq \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$ and $\gamma \in \left(0, \frac{v}{\bar{\alpha}}\right]$; (ii) $C_h \in \left(\frac{\bar{\alpha}(\bar{\alpha}-v)}{2}, \frac{\bar{\alpha}^2}{2}\right)$ and $\gamma \in \left(0, 1 - \frac{2C_h}{\bar{\alpha}^2}\right)$.

3. When only constraint (1) is binding ($\lambda_1 \geq 0$ and $\lambda_2 = 0$): This is the case with $Q_h(p_h) = Q_s(p_s) < \bar{\alpha}$. This constraint gives the following relationship between p_s and p_h:

$$p_s = \frac{(1-\gamma)(p_h - v)}{\gamma}.$$

   Substituting this into the first-order conditions, we get $\lambda_1 = -\frac{v}{\gamma} < 0$ for any γ. Thus this scenario is not supported.

4. When both constraints are binding ($\lambda_1 \geq 0$ and $\lambda_2 \geq 0$): This is the case with $Q_h(p_h) = Q_s(p_s) = \bar{\alpha}$. From the discussion above, we can obtain

$$p_h = v, \quad p_s = 0$$
$$Q_h = Q_s = \bar{\alpha}.$$

Constraint (1) requires $\lambda_1 \geq 0$. However, substituting the above optimal prices into the first-order conditions, we can show that $\lambda_1 = -\bar{\alpha}(1-\gamma) < 0$. Thus, this scenario is not supported.

Combining the results, we have:

1. When $C_h \leq \frac{\bar{\alpha}(\bar{\alpha}-2v)}{2}$, the hardware firm will develop software for $\gamma \in (0,1)$.
   - For $\gamma \in \left(0, \frac{v}{\bar{\alpha}}\right]$, the optimal strategy is characterized by scenario 2.
   - For $\gamma \in \left(\frac{v}{\bar{\alpha}}, 1\right)$, the optimal strategy is characterized by scenario 1.
2. When $C_h \in \left(\frac{\bar{\alpha}(\bar{\alpha}-2v)}{2}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}\right]$, the hardware firm will develop software if $\gamma \in \left(0, \frac{v^2}{2C_h-\bar{\alpha}(\bar{\alpha}-2v)}\right]$. Otherwise, it will only sell hardware.
   - For $\gamma \in \left(0, \frac{v}{\bar{\alpha}}\right]$, the optimal strategy is characterized by scenario 2.
   - For $\gamma \in \left(\frac{v}{\bar{\alpha}}, \frac{v^2}{2C_h-\bar{\alpha}(\bar{\alpha}-2v)}\right)$, the optimal strategy is characterized by scenario 1.
3. When $C_h \in \left(\frac{\bar{\alpha}(\bar{\alpha}-v)}{2}, \frac{\bar{\alpha}^2}{2}\right)$, the hardware firm will develop software if $\gamma \in \left(0, 1-\frac{2C_h}{\bar{\alpha}^2}\right)$. Otherwise, it will only sell hardware.
   - The optimal strategy is characterized by scenario 2.
4. When $C_h \geq \frac{\bar{\alpha}^2}{2}$, the hardware firm will not develop software for $\gamma \in (0,1)$.

In summary, when C_h is sufficiently low, regardless of γ's value, software is supplied. When γ is small, all consumers buy hardware. When C_h is intermediate, the hardware firm does not supply software if γ is large. This is because the negative effect of software piracy on in-house software is so large that it would not make sense to pay C_h and sell software. When C_h is large, it is too costly to supply software so the hardware firm just sells hardware.

A.2 Proof of Proposition 2.3

The software provider's problem given the license fee f is given by

$$\pi_s(f) = \max_{p_s} \, (p_s - f)\left(\bar{\alpha} - \frac{p_s}{1-\gamma}\right) - \frac{C_s}{2}.$$

The first-order condition with respect to p_s yields

$$\begin{aligned} p_s(f) &= \frac{\bar{\alpha}(1-\gamma)+f}{2} \\ Q_s(f) &\equiv \bar{\alpha} - \frac{p_s(f)}{1-\gamma} = \frac{\bar{\alpha}}{2} - \frac{f}{2(1-\gamma)} \\ \pi_s(f) &= \frac{(\bar{\alpha}(1-\gamma)-f)^2}{4(1-\gamma)} - \frac{C_s}{2}. \end{aligned}$$

The hardware provider's problem is then given by

$$\max_{p_h, f} \; p_h Q_h(p_h) + f Q_s(f)$$

subject to (1) $Q_h(p_h) \geq Q_s(f)$, (2) $Q_h(p_h) \leq \bar{\alpha}$, and (3) $\pi_s(f) \geq 0$. The Lagrangian is

$$L(p_h, f, \lambda) = p_h Q_h(p_h) + f Q_s(f) + \lambda_1\big(Q_h(p_h) - Q_s(f)\big) + \lambda_2\big(\bar{\alpha} - Q_h(p_h)\big) + \lambda_3 \pi_s(f).$$

The Kuhn-Tucker conditions are

$$\begin{aligned} \frac{\partial L}{\partial p_h} &= Q_h(p_h) + p_h Q_h'(p_h) + \lambda_1 Q_h'(p_h) - \lambda_2 Q_h'(p_h) = 0 \\ \frac{\partial L}{\partial f} &= Q_s(f) + f Q_s'(f) - \lambda_1 Q_s'(f) + \lambda_3 \pi_s'(f) = 0 \\ Q_h(p_h) \geq Q_s(f) &\quad \lambda_1 \geq 0 \quad \lambda_1\big(Q_h(p_h) - Q_s(f)\big) = 0 \\ Q_h(p_h) \leq \bar{\alpha} &\quad \lambda_2 \geq 0 \quad \lambda_2\big(\bar{\alpha} - Q_h(p_h)\big) = 0 \\ \pi_s(f) \geq 0 &\quad \lambda_3 \geq 0 \quad \lambda_3 \pi_s(f) = 0 \end{aligned}$$

We solve this set of inequalities and equations. Below we examine every possible case.

1. When all constraints are not binding ($\lambda_1 = \lambda_2 = \lambda_3 = 0$): This is the case with interior solutions. We have

$$\begin{aligned} p_h &= \frac{\bar{\alpha}\gamma + v}{2}, \quad & f &= \frac{\bar{\alpha}(1-\gamma)}{2} \\ Q_h &= \frac{\bar{\alpha}\gamma + v}{2\gamma}, \quad & Q_s &= \frac{\bar{\alpha}}{4}. \end{aligned}$$

   Constraint (1) is satisfied for any γ. Non-binding constraint (2) implies that $\gamma \geq \frac{v}{\bar{\alpha}}$. Finally, non-binding constraint (3) implies that $\gamma \leq 1 - \frac{8C_s}{\bar{\alpha}^2}$. The range of γ that supports this scenario is non-empty if $\frac{v}{\bar{\alpha}} \leq 1 - \frac{8C_s}{\bar{\alpha}^2}$ or $C_s \leq \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$.

   With the optimal p_h and f, the hardware firm's profit is

$$\pi_h = \frac{(\bar{\alpha}\gamma + v)^2}{4\gamma} + \frac{\bar{\alpha}^2(1-\gamma)}{8},$$

and it is easy to check that $\pi_h > \bar{\alpha}v$ for any γ in the range. Thus, the hardware firm prefers to have software developed. Moreover,

$$\frac{\partial \pi_h}{\partial \gamma} = \frac{\bar{\alpha}^2}{8} - \frac{v^2}{4\gamma^2}.$$

Thus, $\frac{\partial \pi_h}{\partial \gamma} > 0$ for $\gamma > \frac{\sqrt{2}v}{\bar{\alpha}}$.

In summary, this scenario is supported under the following conditions: $C_h < \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$ and $\gamma \in \left[\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right]$. Moreover, if $C_h < \frac{\bar{\alpha}(\bar{\alpha}-\sqrt{2}v)}{8}$, then $\frac{\partial \pi_h}{\partial \gamma} \leq 0$ for $\gamma \in \left[\frac{v}{\bar{\alpha}}, \frac{\sqrt{2}v}{\bar{\alpha}}\right]$ and $\frac{\partial \pi_h}{\partial \gamma} > 0$ for $\gamma \in \left(\frac{\sqrt{2}v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right]$. If $C_h \in \left[\frac{\bar{\alpha}(\bar{\alpha}-\sqrt{2}v)}{8}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}\right)$, $\frac{\partial \pi_h}{\partial \gamma} \leq 0$ for all $\gamma \in \left[\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right]$.

2. When only constraint (3) is binding ($\lambda_1 = \lambda_2 = 0$ and $\lambda_3 \geq 0$): This is the case where $\pi_s(f) = 0$. We have

$$p_h = \frac{\bar{\alpha}\gamma + v}{2}, \qquad f = \bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}$$
$$Q_h = \frac{\bar{\alpha}\gamma + v}{2\gamma}, \qquad Q_s = \sqrt{\frac{C_s}{2(1-\gamma)}}.$$

Constraint (1) requires $\frac{\bar{\alpha}\gamma+v}{2\gamma} \geq \sqrt{\frac{C_s}{2(1-\gamma)}}$. Let $g(\gamma) = \frac{\bar{\alpha}\gamma+v}{2\gamma} - \sqrt{\frac{C_s}{2(1-\gamma)}}$. It is easy to check that $\frac{\partial g(\gamma)}{\partial \gamma} < 0 \ \forall \gamma$, $\lim_{\gamma \to 0} g(\gamma) = +\infty$, and $\lim_{\gamma \to 1} g(\gamma) = -\infty$. Thus, there exist a unique threshold, say, $\gamma_1 \in (0,1)$ such that $g(\gamma) \geq 0$ for all $\gamma \leq \gamma_1$. Constraint (2) implies that $\gamma \geq \frac{v}{\bar{\alpha}}$. Constraint (3) is binding thus we need $\lambda_3 \geq 0$, which is equivalent to $\gamma \geq 1 - \frac{8C_s}{\bar{\alpha}^2}$. The range of γ that supports this scenario is non-empty if $g(\gamma) > 0$ at $\gamma = \max\left\{\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right\}$. Suppose that $\frac{v}{\bar{\alpha}} < 1 - \frac{8C_s}{\bar{\alpha}^2}$ (or $C_s < \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$). Then, it is easy to check $g(\gamma = 1 - \frac{8C_s}{\bar{\alpha}^2}) > 0$, so $\gamma_1 > 1 - \frac{8C_s}{\bar{\alpha}^2}$. If $\frac{v}{\bar{\alpha}} \geq 1 - \frac{8C_s}{\bar{\alpha}^2}$ (or $C_s \geq \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$), then $g(\gamma = \frac{v}{\bar{\alpha}}) \geq 0$ (i.e., $\frac{v}{\bar{\alpha}} \leq \gamma_1$) if $C_s \leq 2\bar{\alpha}(\bar{\alpha}-v)$. Together, the range of γ is non-empty if $C_s \leq 2\bar{\alpha}(\bar{\alpha}-v)$.

With the optimal p_h and f, the hardware firm's profit is

$$\pi_h = \frac{(\bar{\alpha}\gamma + v)^2}{4\gamma} + \left(\bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}\right)\sqrt{\frac{C_s}{2(1-\gamma)}}.$$

When $\gamma \geq \frac{v}{\bar{\alpha}}$, not everyone buys hardware and $p_h Q_h = \frac{(\bar{\alpha}\gamma+v)^2}{4\gamma} \geq \bar{\alpha}v$. Thus, the above equation for π_h suggests that π_h can be lower than $\bar{\alpha}v$ only when the optimal licensing fee is negative:

$$\bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s} < 0 \Leftrightarrow \gamma > 1 - \frac{2C_s}{\bar{\alpha}^2}.$$

We first check if a negative licensing fee could actually happen within the above range of γ. When $\frac{v}{\bar{\alpha}} > 1 - \frac{2C_s}{\bar{\alpha}^2}$ (or $C_s > \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$), then the condition for this scenario to be non-empty

$(C_s \le 2\bar{\alpha}(\bar{\alpha}-v))$ implies that $1-\frac{2C_s}{\bar{\alpha}^2} < \gamma_1$. If $\frac{v}{\bar{\alpha}} \le 1-\frac{2C_s}{\bar{\alpha}^2}$, then it can be shown that $g(\gamma = 1-\frac{2C_s}{\bar{\alpha}^2}) = \frac{\bar{\alpha}^2 v}{2(\bar{\alpha}^2-2C_s)} > 0$. Thus, the negative licensing fee could indeed happen. Below, we focus on $\gamma \in \big(1-\frac{2C_s}{\bar{\alpha}^2},\gamma_1\big]$.

First, note that

$$\frac{\partial \pi_h}{\partial \gamma} = \frac{\bar{\alpha}^2}{4} - \frac{v^2}{4\gamma^2} - \frac{\bar{\alpha}}{2}\sqrt{\frac{C_s}{2(1-\gamma)}}.$$

We can show that for $\gamma > 1-\frac{2C_s}{\bar{\alpha}^2}$, $\sqrt{\frac{C_s}{2(1-\gamma)}} > \frac{\bar{\alpha}}{2}$. Thus, we have $\frac{\partial \pi_h}{\partial \gamma} < 0$ for $\gamma \in \big(1-\frac{2C_s}{\bar{\alpha}^2},\gamma_1\big]$. Given this, if $1-\frac{2C_s}{\bar{\alpha}^2} < \frac{v}{\bar{\alpha}}$ (or $C_s > \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$), then we have $\pi_h < \bar{\alpha}v$ for $\gamma \in \big[\frac{v}{\bar{\alpha}},\gamma_1\big]$ because $\pi_h < \bar{\alpha}v$ at $\gamma = \frac{v}{\bar{\alpha}}$. For $C_s \le \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$, since $\pi > \bar{\alpha}v$ at $\gamma = 1-\frac{2C_s}{\bar{\alpha}^2}$, we can possibly have a unique threshold,say, $\gamma_2 \in \big(1-\frac{2C_s}{\bar{\alpha}^2},\gamma_2\big]$ such that $\pi_h < \bar{\alpha}v$ for $\gamma \in (\gamma_2,\gamma_1]$. For example, we can numerically check that for $(\bar{\alpha},v,C_s) = (1.0,0.5,0.2)$, there exists such $\gamma_2 < \gamma_1$, but for $(\bar{\alpha},v,C_s) = (1.5,0.5,0.2)$, $\pi_h > \bar{\alpha}v$ at $\gamma = \gamma_1$. Since we cannot derive a closed form for the thresholds, let us define $\Theta \equiv \{\theta = (\bar{\alpha},v,C_s) : 0 < v < \bar{\alpha}, 0 < C_s < \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}, \pi_h(\gamma=\gamma_1) \ge \bar{\alpha}v\}$.

We can now summarize the range of γ that supports this scenario. When $C_h < \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$, $\gamma \in \big[1-\frac{8C_s}{\bar{\alpha}^2},\gamma_1\big]$ for $(\bar{\alpha},v,C_s) \in \Theta$. For $(\bar{\alpha},v,C_s) \notin \Theta$, there exists a unique $\gamma_2(\theta) \in \big(1-\frac{2C_s}{\bar{\alpha}^2},\gamma_1\big)$ such that $\pi_h < \bar{\alpha}v$ for all $\gamma \in (\gamma_2(\theta),\gamma_1]$. When $C_h \in \Big[\frac{\bar{\alpha}(\bar{\alpha}-v)}{8},\frac{\bar{\alpha}(\bar{\alpha}-v)}{2}\Big]$, $\gamma \in \big[\frac{v}{\bar{\alpha}},\gamma_1\big]$ for $(\bar{\alpha},v,C_s) \in \Theta$. For $(\bar{\alpha},v,C_s) \notin \Theta$, there exists a unique $\gamma_2(\theta) \in \big(1-\frac{2C_s}{\bar{\alpha}^2},\gamma_1\big)$ such that $\pi_h < \bar{\alpha}v$ for all $\gamma \in (\gamma_2(\theta),\gamma_1]$. When $C_h \in \Big(\frac{\bar{\alpha}(\bar{\alpha}-v)}{2},2\bar{\alpha}(\bar{\alpha}-v)\Big]$, $\gamma \in \big[\frac{v}{\bar{\alpha}},\gamma_1\big]$. But as we saw, $\pi_h < \bar{\alpha}v$ for this range of γ and the hardware firm prefers not to have software.

Finally, we examine the effect of γ on π_h. We saw that for $\gamma \in \big(1-\frac{2C_s}{\bar{\alpha}^2},\gamma_1\big]$, we have $\frac{\partial \pi_h}{\partial \gamma} < 0$. We thus consider $\gamma \in \big[\max\big\{\frac{v}{\bar{\alpha}},1-\frac{8C_s}{\bar{\alpha}^2}\big\},1-\frac{2C_s}{\bar{\alpha}^2}\big]$. Note that

$$\frac{\partial^2 \pi_h}{\partial^2 \gamma} = \frac{v^2}{2\gamma^3} - \frac{\bar{\alpha}}{4(1-\gamma)}\sqrt{\frac{C_s}{2(1-\gamma)}} = \frac{v^2}{2(1-\gamma)\sqrt{1-\gamma}}\left(\frac{(1-\gamma)\sqrt{1-\gamma}}{\gamma^3} - \frac{\bar{\alpha}\sqrt{C_s}}{2\sqrt{2}v^2}\right).$$

It is easy to check that $\lim_{\gamma\to 0}\frac{\partial^2 \pi_h}{\partial^2 \gamma} = \infty$ and $\lim_{\gamma\to 1}\frac{\partial^2 \pi_h}{\partial^2 \gamma} = -\infty$. Let $h(\gamma) = \frac{(1-\gamma)\sqrt{1-\gamma}}{\gamma^3} - \frac{\bar{\alpha}\sqrt{C_s}}{2\sqrt{2}v^2}$. We can show that $\frac{\partial h(\gamma)}{\partial \gamma} < 0$ for all γ, and thus, there exists a unique threshold, say, γ_3 such that $\frac{\partial^2 \pi_h}{\partial^2 \gamma}\Big|_{\gamma=\gamma_3} = 0$. We can then consider three possibilities in terms of where γ_3 falls into the range of γ above. Suppose $\gamma_3 > 1-\frac{2C_s}{\bar{\alpha}^2}$. Then $\frac{\partial^2 \pi_h}{\partial^2 \gamma} > 0$ for $\gamma \in \big[\max\big\{\frac{v}{\bar{\alpha}},1-\frac{8C_s}{\bar{\alpha}^2}\big\},1-\frac{2C_s}{\bar{\alpha}^2}\big]$. Since $\frac{\partial \pi_h}{\partial \gamma} < 0$ at $\gamma = 1-\frac{2C_s}{\bar{\alpha}^2}$, we have $\frac{\partial \pi_h}{\partial \gamma} < 0$ for $\forall \gamma \in \big[\max\big\{\frac{v}{\bar{\alpha}},1-\frac{8C_s}{\bar{\alpha}^2}\big\},1-\frac{2C_s}{\bar{\alpha}^2}\big]$. For this to happen, we need $h(\gamma = 1-\frac{2C_s}{\bar{\alpha}^2}) > 0$. Next, if $\gamma_3 < \max\big\{\frac{v}{\bar{\alpha}},1-\frac{8C_s}{\bar{\alpha}^2}\big\}$, then $\frac{\partial^2 \pi_h}{\partial^2 \gamma} < 0$ for $\gamma \in \big[\max\big\{\frac{v}{\bar{\alpha}},1-\frac{8C_s}{\bar{\alpha}^2}\big\},1-\frac{2C_s}{\bar{\alpha}^2}\big]$. Thus the sign of $\frac{\partial \pi_h}{\partial \gamma}$ depends on whether $\frac{\partial \pi_h}{\partial \gamma}$ is positive or negative at $\gamma = \max\big\{\frac{v}{\bar{\alpha}},1-\frac{8C_s}{\bar{\alpha}^2}\big\}$. First, when

$\frac{v}{\bar{\alpha}} < 1 - \frac{8C_s}{\bar{\alpha}^2}$ (or $C_s < \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$), we have

$$\left.\frac{\partial \pi_h}{\partial \gamma}\right|_{\gamma=1-\frac{8C_s}{\bar{\alpha}^2}} = \frac{\bar{\alpha}^2}{4}\left[\frac{1}{2} - \left(\frac{\bar{\alpha}v}{\bar{\alpha}^2 - 8C_s}\right)^2\right],$$

which is positive if $C_s < \frac{\bar{\alpha}(\bar{\alpha}-\sqrt{2}v)}{8}$. Thus, when $C_s < \frac{\bar{\alpha}(\bar{\alpha}-\sqrt{2}v)}{8}$, there exists a unique threshold, say, γ_4 such that $\frac{\partial \pi_h}{\partial \gamma} > 0$ for $\gamma \in \left(1 - \frac{8C_s}{\bar{\alpha}^2}, \gamma_4\right)$ and $\frac{\partial \pi_h}{\partial \gamma} \leq 0$ otherwise. If $C_s \in \left[\frac{\bar{\alpha}(\bar{\alpha}-\sqrt{2}v)}{8}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}\right)$, then $\frac{\partial \pi_h}{\partial \gamma} < 0$ for all $\gamma \in \left(1 - \frac{8C_s}{\bar{\alpha}^2}, 1 - \frac{2C_s}{\bar{\alpha}^2}\right)$. When $\frac{v}{\bar{\alpha}} < 1 - \frac{8C_s}{\bar{\alpha}^2}$, it is easy to check

$$\left.\frac{\partial \pi_h}{\partial \gamma}\right|_{\gamma=\frac{v}{\bar{\alpha}}} = -\frac{\bar{\alpha}}{2}\sqrt{\frac{\bar{\alpha}C_s}{2(\bar{\alpha}-v)}} < 0 \quad \forall \gamma \in \left(\frac{v}{\bar{\alpha}}, 1 - \frac{2C_s}{\bar{\alpha}^2}\right).$$

Finally, if $\gamma_3 \gamma \in \left[\max\left\{\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right\}, 1 - \frac{2C_s}{\bar{\alpha}^2}\right]$, then $\frac{\partial \pi_h}{\partial \gamma}$ is a parabola with a maximum at $\gamma = \gamma_3$. Thus, we can check if the maximum attained can be positive. Analytically, it is cumbersome to show. However, our numerical analysis shows that there exists a set of $(\bar{\alpha}, v, C_s)$ such that the maximum is positive (e.g., $(\bar{\alpha}, v, C_s) = (1, 0.2, 0.1)$). Under such a condition, there exist $\gamma_4, \gamma_5 \in \left[\max\left\{\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right\}, 1 - \frac{2C_s}{\bar{\alpha}^2}\right]$ such that $\frac{\partial \pi_h}{\partial \gamma} > 0$ for $\gamma \in (\gamma_4, \gamma_5)$ and $\frac{\partial \pi_h}{\partial \gamma} \leq 0$ otherwise.

3. When only constraint (2) is binding ($\lambda_1 = 0$, $\lambda_2 \geq 0$, and $\lambda_3 = 0$): This is the case where $Q_h(p_h) = \bar{\alpha}$. We have

$$p_h = v, \quad f = \frac{\bar{\alpha}(1-\gamma)}{2}$$
$$Q_h = \bar{\alpha}, \quad Q_s = \frac{\bar{\alpha}}{4}.$$

   Constraint (1) is satisfied for any γ. Constraint (2) is binding thus we need $\lambda_2 \geq 0$, which results in $\gamma \leq \frac{v}{\bar{\alpha}}$. Constraint (3) implies $\gamma \leq 1 - \frac{8C_s}{\bar{\alpha}^2}$. The range of γ that supports this scenario is non-empty if $1 - \frac{8C_s}{\bar{\alpha}^2} > 0$, or $C_s < \frac{\bar{\alpha}^2}{8}$.

   The hardware firm's profit is $\pi_h = \bar{\alpha}v + \frac{\bar{\alpha}^2(1-\gamma)}{8}$, which is greater than $\bar{\alpha}v$ for any $\gamma \leq \min\left\{\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\right\}$. Moreover, $\frac{\partial \pi_h}{\partial \gamma} < 0$.

   In summary, this scenario is supported under the following conditions: (i) $C_s \leq \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$ and $\gamma \in \left(0, \frac{v}{\bar{\alpha}}\right]$; (ii) $C_s \in \left[\frac{\bar{\alpha}(\bar{\alpha}-v)}{8}, \frac{\bar{\alpha}^2}{8}\right]$ and $\gamma \in \left(0, 1 - \frac{8C_s}{\bar{\alpha}^2}\right)$. In both cases, $\frac{\partial \pi_h}{\partial \gamma} < 0$.

4. When constraints (2) and (3) are binding ($\lambda_1 = 0$, $\lambda_2 \geq 0$, and $\lambda_3 \geq 0$): This is the case

where $Q_s(f) \le Q_h(p_h) = \bar{\alpha}$ and $\pi_s(f) = 0$. We have

$$p_h = v, \quad f = \bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}$$
$$Q_h = \bar{\alpha}, \quad Q_s = \sqrt{\frac{C_s}{2(1-\gamma)}}.$$

Constraint (1) requires $\bar{\alpha} \ge \sqrt{\frac{C_s}{2(1-\gamma)}}$, which is equivalent to $\gamma \le 1 - \frac{C_s}{2\bar{\alpha}^2}$. Constraint (2) is binding thus we need $\lambda_2 \ge 0$, or $\gamma \le \frac{v}{\bar{\alpha}}$. Constraint (3) is also binding thus we need $\lambda_3 \ge 0$, or $\gamma \ge 1 - \frac{8C_s}{\bar{\alpha}^2}$. The range of γ that supports this scenario is $\gamma \in \left[1 - \frac{8C_s}{\bar{\alpha}^2}, \min\left\{\frac{v}{\bar{\alpha}}, 1 - \frac{C_s}{2\bar{\alpha}^2}\right\}\right]$. In order for this range to be non-empty, we need the following conditions. When $\frac{v}{\bar{\alpha}} \le 1 - \frac{C_s}{2\bar{\alpha}^2}$ (or $C_s \le 2\bar{\alpha}(\bar{\alpha} - v)$), we need $1 - \frac{8C_s}{\bar{\alpha}^2} \le \frac{v}{\bar{\alpha}}$, or $C_s \ge \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$. When $\frac{v}{\bar{\alpha}} > 1 - \frac{C_s}{2\bar{\alpha}^2}$, we need $1 - \frac{C_s}{2\bar{\alpha}^2} > 0$, or $C_s < 2\bar{\alpha}^2$. Thus, the range of γ is non-empty if $C_s \in \left[\frac{\bar{\alpha}(\bar{\alpha}-v)}{8}, 2\bar{\alpha}^2\right)$.

With the optimal p_h and f, the hardware firm's profit is

$$\pi_h = \bar{\alpha}v + \left(\bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}\right)\sqrt{\frac{C_s}{2(1-\gamma)}}.$$

This is greater than or equal to $\bar{\alpha}v$ as long as the licensing fee is non-negative, which implies $\gamma \le 1 - \frac{2C_s}{\bar{\alpha}^2}$. Since $1 - \frac{2C_s}{\bar{\alpha}^2} < 1 - \frac{C_s}{2\bar{\alpha}^2}$, the range of γ that supports this scenario becomes $\gamma \in \left[1 - \frac{8C_s}{\bar{\alpha}^2}, \min\left\{\frac{v}{\bar{\alpha}}, 1 - \frac{2C_s}{\bar{\alpha}^2}\right\}\right]$. When $\frac{v}{\bar{\alpha}} \le 1 - \frac{2C_s}{\bar{\alpha}^2}$ (or $C_s \le \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$), we need $1 - \frac{8C_s}{\bar{\alpha}^2} \le \frac{v}{\bar{\alpha}}$, or $C_s \ge \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$. When $\frac{v}{\bar{\alpha}} > 1 - \frac{2C_s}{\bar{\alpha}^2}$, we need $1 - \frac{2C_s}{\bar{\alpha}^2} > 0$, or $C_s < \frac{\bar{\alpha}^2}{2}$. Thus, the range of γ is non-empty if $C_s \in \left[\frac{\bar{\alpha}(\bar{\alpha}-v)}{8}, \frac{\bar{\alpha}^2}{2}\right)$. Moreover, note that

$$\frac{\partial \pi_h}{\partial \gamma} = -\frac{\bar{\alpha}}{2}\sqrt{\frac{C_s}{2(1-\gamma)}}.$$

Thus $\frac{\partial \pi_h}{\partial \gamma} < 0$.

In summary, this scenario is supported under the following conditions: (i) $C_s \in \left[\frac{\bar{\alpha}(\bar{\alpha}-v)}{8}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}\right]$ and $\gamma \in \left(\max\left\{0, 1 - \frac{8C_s}{\bar{\alpha}^2}\right\}, \frac{v}{\bar{\alpha}}\right]$; (ii) $C_s \in \left(\frac{\bar{\alpha}(\bar{\alpha}-v)}{2}, \frac{\bar{\alpha}^2}{2}\right]$, $\gamma \in \left(\max\left\{0, 1 - \frac{8C_s}{\bar{\alpha}^2}\right\}, 1 - \frac{2C_s}{\bar{\alpha}^2}\right]$. For $C_s \in \left(\frac{\bar{\alpha}^2}{2}, 2\bar{\alpha}^2\right]$, the optimal strategy is supported, but the hardware firm's profits are lower than $\bar{\alpha}v$. Moreover, we have $\frac{\partial \pi_h}{\partial \gamma} < 0$ under these conditions.

5. When only constraint (1) is binding ($\lambda_1 \ge 0$ and $\lambda_2 = \lambda_3 = 0$): This is the case with $Q_h(p_h) = Q_s(f)$. Intuitively, this scenario will not be supported because when $\pi_s(f) > 0$, the optimal licensing fee will be high enough to make $Q_s(f)$ smaller than $Q_h(p_h)$. We can compute

$$p_h = \frac{\bar{\alpha}(3-2\gamma)\gamma + (4-3\gamma)v}{2(2-\gamma)}, \quad f = \frac{(\bar{\alpha}(1-\gamma) - v)(1-\gamma)}{2-\gamma}.$$

The first-order condition with respect to f gives $\lambda_1 = \frac{f}{\gamma(1-\gamma)} - \frac{\bar{\alpha}}{2\gamma}$. Substituting the optimal f into this, we can show that $\lambda_1 < 0$ for any γ. Thus, this scenario is not supported.

6. When constraints (1) and (3) are binding ($\lambda_1 \geq 0$, $\lambda_2 = 0$, and $\lambda_3 \geq 0$): This is the case with $Q_h(p_h) = Q_s(f) < \bar{\alpha}$ and $\pi_s(f) = 0$. This could happen when C_s is large and γ is also large so that the hardware firm needs to lower f sufficiently, which makes the software demand equal to the hardware demand. We can compute

$$p_h = \bar{\alpha}\gamma + v - \gamma\sqrt{\frac{C_s}{2(1-\gamma)}}, \qquad f = \bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}$$
$$Q_h = Q_s = \sqrt{\frac{C_s}{2(1-\gamma)}}.$$

Constraint (1) is binding thus we need $\lambda_1 \geq 0$. From the first-order condition with respect to p_h, we get

$$\lambda_1 = -\bar{\alpha}\gamma - v + 2\gamma\sqrt{\frac{C_s}{2(1-\gamma)}} = 2\gamma\left(\sqrt{\frac{C_s}{2(1-\gamma)}} - \frac{\bar{\alpha}\gamma + v}{2\gamma}\right),$$

which is greater than or equal to zero when $\gamma \geq \gamma_1$, where $\tilde{\gamma}$ is defined in scenario 2 above. Constraint (2) requires $Q_h \leq \bar{\alpha}$, or $\gamma \leq 1 - \frac{C_s}{2\bar{\alpha}^2}$. Constraint (3) is binding thus we need $\lambda_3 \geq 0$, which can be shown to be satisfied when $\gamma \geq \gamma_1$. To see this, note that the first-order condition with respect to f gives

$$\lambda_3 \geq 0 \Leftrightarrow \bar{\alpha}(1-\gamma) - 2f + \lambda_1 \geq 0.$$

First, note that $\bar{\alpha}(1-\gamma) - 2f \geq 0$ for $\gamma \geq 1 - \frac{8C_s}{\bar{\alpha}^2}$ and that $\gamma_1 > 1 - \frac{8C_s}{\bar{\alpha}^2}$ (we have proven this in scenario 2 above, but it is easy to check that because $\lambda_1(\gamma = 1 - \frac{8C_s}{\bar{\alpha}^2}) < 0$ and $\frac{\partial \lambda_1}{\partial \gamma} > 0$, it must be $1 - \frac{8C_s}{\bar{\alpha}^2} < \gamma_1$). Thus, at $\gamma = \gamma_1$, $\lambda_3 > 0$. Overall, the range of γ that supports this scenario is $\gamma \in \left[\gamma_1, 1 - \frac{C_s}{2\bar{\alpha}^2}\right]$. For this range of γ to be non-empty, we need $\lambda_1(\gamma = 1 - \frac{C_s}{2\bar{\alpha}^2}) \leq 0$, which is equivalent to $C_s \leq 2\bar{\alpha}(\bar{\alpha} - v)$.

With the optimal p_h and f, the hardware firm's profit is

$$\begin{aligned}\pi_h &= \left(\bar{\alpha}\gamma + v - \gamma\sqrt{\frac{C_s}{2(1-\gamma)}}\right)\sqrt{\frac{C_s}{2(1-\gamma)}} + \left(\bar{\alpha}(1-\gamma) - \sqrt{2(1-\gamma)C_s}\right)\sqrt{\frac{C_s}{2(1-\gamma)}} \\ &= \left(\bar{\alpha} + v - (2-\gamma)\sqrt{\frac{C_s}{2(1-\gamma)}}\right)\sqrt{\frac{C_s}{2(1-\gamma)}}.\end{aligned}$$

We note that at $\gamma = 1 - \frac{C_s}{2\bar{\alpha}^2}$, we have $p_h = v$, $f = -\frac{C_s}{2\bar{\alpha}}$, $Q_h = Q_s = \bar{\alpha}$. Thus,

$$\pi_h = \left(v - \frac{C_s}{2\bar{\alpha}}\right)\bar{\alpha} = \bar{\alpha}v - \frac{C_s}{2},$$

which is clearly less than $\bar{\alpha}v$. For $\gamma \in \left[\gamma_1, 1 - \frac{C_s}{2\bar{\alpha}}\right)$, note that

$$\frac{\partial \pi_h}{\partial \gamma} = \frac{1}{1-\gamma}\sqrt{\frac{C_s}{2(1-\gamma)}}\left(\frac{\bar{\alpha}+v}{2} - \sqrt{\frac{C_s}{2(1-\gamma)}}\right).$$

We know that $\frac{\bar{\alpha}+v}{2} < \frac{\bar{\alpha}\gamma+v}{2\gamma}$ for $\gamma < 1$. Also, for $\gamma > \gamma_1$, $\frac{\bar{\alpha}\gamma+v}{2\gamma} < \sqrt{\frac{C_s}{2(1-\gamma)}}$. Thus, $\frac{\partial \pi_h}{\partial \gamma} < 0$ for $\gamma \in \left[\gamma_1, 1 - \frac{C_s}{2\bar{\alpha}^2}\right)$. As we saw in scenario 2, we can show that $\pi_h - \bar{\alpha}v$ at $\gamma = \gamma_1$ can be positive or negative under some values of $(\bar{\alpha}, v, C_s)$. Note that the hardware firm's profit function in this scenario is identical to that in scenario 2 at $\gamma = \gamma_1$. Thus, we can use the same set of Θ we defined in scenario 2, and summarize the results as follows.

When $C_h < \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$ and $(\bar{\alpha}, v, C_s) \in \Theta$, there exists a unique threshold, say, $\gamma_6(\theta) \in \left(\gamma_1, 1 - \frac{C_s}{2\bar{\alpha}^2}\right)$ such that the hardware firm prefers to have software for $\gamma \in [\gamma_1, \gamma_6(\theta)]$. When $C_h < \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}$ and $(\bar{\alpha}, v, C_s) \notin \Theta$ or $C_h \in \left[\frac{\bar{\alpha}(\bar{\alpha}-v)}{2}, 2\bar{\alpha}(\bar{\alpha}-v)\right]$, the hardware firm prefers not to have software. Moreover, $\frac{\partial \pi_h}{\partial \gamma} < 0$.

7. When constraints (1) and (2) are binding ($\lambda_1 \geq 0$, $\lambda_2 \geq 0$, and $\lambda_3 = 0$): This is the case with $Q_h(p_h) = Q_s(f) = \bar{\alpha}$. Once again, intuitively, this scenario will not be supported because when $\pi_s(f) > 0$, the optimal licensing fee will be high enough to make $Q_s(f)$ smaller than $Q_h(p_h)$. We can compute

$$p_h = v \qquad f = -\bar{\alpha}(1-\gamma).$$

The first-order condition with respect to f gives $\lambda_1 = \frac{f}{\gamma(1-\gamma)} - \frac{\bar{\alpha}}{2\gamma}$. Substituting the optimal f into this, we can show that $\lambda_1 < 0$ for any γ.

8. When all constraints are binding ($\lambda_1 \geq 0$, $\lambda_2 \geq 0$, and $\lambda_3 \geq 0$): This is the case with $Q_h(p_h) = Q_s(f) = \bar{\alpha}$ and $\pi_s(f) = 0$. We can compute

$$p_h = v \qquad f = -\bar{\alpha}(1-\gamma)$$
$$Q_h = Q_s = \bar{\alpha}$$

From the three constraints, we can show that this scenario is supported only at $\gamma = 1 - \frac{C_s}{2\bar{\alpha}^2}$. At this γ, the first-order conditions give

$$\lambda_3 = \frac{\bar{\alpha}}{C_s}\lambda_1 + 1, \qquad \lambda_2 = \lambda_1 + v + \frac{C_s}{2\bar{\alpha}} - \bar{\alpha}.$$

First, for any $\lambda_1 \geq$, $\lambda_3 > 0$, but $\lambda_2 \geq 0$ if $v + \frac{C_s}{2\bar{\alpha}} - \bar{\alpha} \geq 0$, or $C_s \geq 2\bar{\alpha}(\bar{\alpha}-v)$.

Note that the hardware firm's profits under the above optimal strategy are then

$$\pi_h = \bar{\alpha}v - \bar{\alpha}(1-\gamma)\bar{\alpha} < \bar{\alpha}v.$$

Thus, the hardware firm prefers not to have software.

Combining the results from supported scenarios, we have:

1. When $C_s \leq \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$ and $\theta = (\bar{\alpha}, v, C_s) \in \Theta$, there exists a unique $\gamma_6(\theta) \in \big(\gamma_1, 1 - \frac{C_s}{2\bar{\alpha}^2}\big)$ such that the optimal strategy is characterized as follows.
    - For $\gamma \in \big(0, \frac{v}{\bar{\alpha}}\big]$, the optimal strategy is characterized by scenario 3.
    - For $\gamma \in \big(\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\big)$, the optimal strategy is characterized by scenario 1.
    - For $\gamma \in \big(1 - \frac{8C_s}{\bar{\alpha}^2}, \gamma_1\big)$, the optimal strategy is characterized by scenario 2.
    - For $\gamma \in (\gamma_1, \gamma_6(\theta)]$, the optimal strategy is characterized by scenario 6.
    - For $\gamma \in (\gamma_6(\theta), 1)$, the hardware firm chooses not to have software.
2. When $C_s \leq \frac{\bar{\alpha}(\bar{\alpha}-v)}{8}$ and $\theta = (\bar{\alpha}, v, C_s) \notin \Theta$, there exists a unique $\gamma_2(\theta) \in \big(1 - \frac{2C_s}{\bar{\alpha}^2}, \gamma_1\big)$ such that the optimal strategy is characterized as follows.
    - For $\gamma \in \big(0, \frac{v}{\bar{\alpha}}\big]$, the optimal strategy is characterized by scenario 3.
    - For $\gamma \in \big(\frac{v}{\bar{\alpha}}, 1 - \frac{8C_s}{\bar{\alpha}^2}\big)$, the optimal strategy is characterized by scenario 1.
    - For $\gamma \in \big(1 - \frac{8C_s}{\bar{\alpha}^2}, \gamma_2(\theta)\big]$, the optimal strategy is characterized by scenario 2.
    - For $\gamma \in (\gamma_2(\theta), 1)$, the hardware firm chooses not to have software.
3. When $C_s \in \Big(\frac{\bar{\alpha}(\bar{\alpha}-v)}{8}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}\Big]$ and $\theta = (\bar{\alpha}, v, C_s) \in \Theta$, there exists a unique $\gamma_6(\theta) \in \big(\gamma_1, 1 - \frac{C_s}{2\bar{\alpha}^2}\big)$ such that the optimal strategy is characterized as follows.
    - For $C_s \leq \frac{\bar{\alpha}^2}{8}$:
        - For $\gamma \in \big(0, 1 - \frac{8C_s}{\bar{\alpha}^2}\big]$, the optimal strategy is characterized by scenario 3.
        - For $\gamma \in \big(1 - \frac{8C_s}{\bar{\alpha}^2}, \frac{v}{\bar{\alpha}}\big)$, the optimal strategy is characterized by scenario 4.
        - For $\gamma \in \big(\frac{v}{\bar{\alpha}}, \gamma_1\big)$, the optimal strategy is characterized by scenario 2.
        - For $\gamma \in (\gamma_1, \gamma_6(\theta)]$, the optimal strategy is characterized by scenario 6.
        - For $\gamma \in (\gamma_6(\theta), 1)$, the hardware firm chooses not to have software.
    - For $C_s > \frac{\bar{\alpha}^2}{8}$:

  - For $\gamma \in \left(0, \frac{v}{\bar{\alpha}}\right)$, the optimal strategy is characterized by scenario 4.
  - For $\gamma \in \left(\frac{v}{\bar{\alpha}}, \gamma_1\right)$, the optimal strategy is characterized by scenario 2.
  - For $\gamma \in (\gamma_1, \gamma_6(\theta)]$, the optimal strategy is characterized by scenario 6.
  - For $\gamma \in (\gamma_6(\theta), 1)$, the hardware firm chooses not to have software.

4. When $C_s \in \left(\frac{\bar{\alpha}(\bar{\alpha}-v)}{8}, \frac{\bar{\alpha}(\bar{\alpha}-v)}{2}\right]$ and $\theta = (\bar{\alpha}, v, C_s) \notin \Theta$, there exists a unique $\gamma_2(\theta) \in \left(1 - \frac{2C_s}{\bar{\alpha}^2}, \gamma_1\right)$ such that the optimal strategy is characterized as follows.

   - For $C_s \leq \frac{\bar{\alpha}^2}{8}$:
     - For $\gamma \in \left(0, 1 - \frac{8C_s}{\bar{\alpha}^2}\right]$, the optimal strategy is characterized by scenario 3.
     - For $\gamma \in \left(1 - \frac{8C_s}{\bar{\alpha}^2}, \frac{v}{\bar{\alpha}}\right)$, the optimal strategy is characterized by scenario 4.
     - For $\gamma \in \left(\frac{v}{\bar{\alpha}}, \gamma_2(\theta)\right]$, the optimal strategy is characterized by scenario 2.
     - For $\gamma \in (\gamma_2(\theta), 1)$, the hardware firm chooses not to have software.
   - For $C_s > \frac{\bar{\alpha}^2}{8}$:
     - For $\gamma \in \left(0, \frac{v}{\bar{\alpha}}\right)$, the optimal strategy is characterized by scenario 4.
     - For $\gamma \in \left(\frac{v}{\bar{\alpha}}, \gamma_2(\theta)\right]$, the optimal strategy is characterized by scenario 2.
     - For $\gamma \in (\gamma_2(\theta), 1)$, the hardware firm chooses not to have software.

5. When $C_s \in \left(\frac{\bar{\alpha}(\bar{\alpha}-v)}{2}, \frac{\bar{\alpha}^2}{2}\right]$, the optimal strategy is characterized as follows. Note that Assumption 2.1 implies that $\frac{\bar{\alpha}(\bar{\alpha}-v)}{2} > \frac{\bar{\alpha}^2}{8}$, thus $C_s > \frac{\bar{\alpha}^2}{8}$ in this region of C_s.

   - For $\gamma \in \left(0, 1 - \frac{2C_s}{\bar{\alpha}^2}\right]$, the optimal strategy is characterized by scenario 4.
   - For $\gamma \in \left(1 - \frac{2C_s}{\bar{\alpha}^2}, 1\right)$, the hardware firm chooses not to have software.

6. When $C_s > \frac{\bar{\alpha}^2}{2}$, the hardware firm prefers not to have software for $\gamma \in (0, 1)$.

A.3 Proof of Proposition 2.6

The software provider's problem given the license fee f is given by

$$\pi_s(f, \delta) = \max_{p_s} \, (1-\delta)(p_s - f)\left(\bar{\alpha} - \frac{p_s}{1-\gamma}\right) - \frac{C_s}{2}(1-\delta)^2.$$

The first-order condition with respect to p_s yields

$$\begin{aligned} p_s(f) &= \frac{\bar{\alpha}(1-\gamma) + f}{2} \\ Q_s(f) \equiv \bar{\alpha} - \frac{p_s(f)}{1-\gamma} &= \frac{\bar{\alpha}}{2} - \frac{f}{2(1-\gamma)} \\ \pi_s(f, \delta) &= (1-\delta)\frac{(\bar{\alpha}(1-\gamma) - f)^2}{4(1-\gamma)} - \frac{C_s}{2}(1-\delta)^2. \end{aligned}$$

The hardware provider's problem is then given by

$$\max_{p_h, f, \delta} \; p_h Q_h(p_h) + \big(\delta p_s(f) + (1-\delta) f\big) Q_s(f) - \frac{C_h}{2}\delta^2$$

subject to (1) $Q_h(p_h) \geq Q_s(f)$, (2) $Q_h(p_h) \leq \bar{\alpha}$, and (3) $\pi_s(f) \geq 0$. The Lagrangian is

$$\begin{aligned} L(p_h, f, \lambda) = {} & p_h Q_h(p_h) + \big(\delta p_s(f) + (1-\delta) f\big) Q_s(f) - \frac{C_h}{2}\delta^2 \\ & + \lambda_1\big(Q_h(p_h) - Q_s(f)\big) + \lambda_2\big(\bar{\alpha} - Q_h(p_h)\big) + \lambda_3 \pi_s(f, \delta). \end{aligned}$$

The Kuhn-Tucker conditions are

$$\begin{aligned} \frac{\partial L}{\partial p_h} &= Q_h(p_h) + p_h Q_h'(p_h) + \lambda_1 Q_h'(p_h) - \lambda_2 Q_h'(p_h) = 0 \\ \frac{\partial L}{\partial f} &= \left(\delta p_s'(f) + (1-\delta)\right) Q_s(f) + \big(\delta p_s(f) + (1-\delta) f\big) Q_s'(f) - \lambda_1 Q_s'(f) + \lambda_3 \pi_{s,f}'(f, \delta) = 0 \\ \frac{\partial L}{\partial \delta} &= (p_s(f) - f) Q_s(f) - C_h \delta + \lambda_3 \pi_{s,\delta}'(f, \delta) = 0 \\ Q_h(p_h) &\geq Q_s(f) \quad \lambda_1 \geq 0 \quad \lambda_1\big(Q_h(p_h) - Q_s(f)\big) = 0 \\ Q_h(p_h) &\leq \bar{\alpha} \quad \lambda_2 \geq 0 \quad \lambda_2\big(\bar{\alpha} - Q_h(p_h)\big) = 0 \\ \pi_s(f, \delta) &\geq 0 \quad \lambda_3 \geq 0 \quad \lambda_3 \pi_s(f, \delta) = 0 \end{aligned}$$

We solve this set of inequalities and equations. Given Assumptions 2.4 and 2.5, we will restrict our attention to the interior solution case ($\lambda_1 = \lambda_2 = \lambda_3 = 0$).

We have

$$\begin{aligned} p_h &= \frac{\bar{\alpha}\gamma + v}{2} & f &= \frac{\bar{\alpha}(1-\gamma)(1-\delta)}{2-\delta} \\ Q_h &= \frac{\bar{\alpha}\gamma + v}{2\gamma} & Q_s &= \frac{\bar{\alpha}}{2(2-\delta)}. \end{aligned}$$

Constraint (1) is satisfied for any γ. Constraint (2) implies that $\gamma \geq \frac{v}{\bar{\alpha}}$. Assumptions 2.4 and 2.5 imply that this condition is satisfied. Finally, Constraint (3) implies that $\delta \geq \frac{C_s}{2C_h + C_s}$. Since $C_h \geq C_s$, we have $\frac{C_s}{2C_h + C_s} \in \left(0, \frac{1}{3}\right]$. The first-order condition gives

$$\delta(2-\delta)^2 = \frac{\bar{\alpha}^2(1-\gamma)}{4C_h}.$$

As in Figure 2, the lower bound of γ, $\underline{\gamma}$, in Assumption 2.5 guarantees that optimal δ exists. When $\frac{\bar{\alpha}^2(1-\gamma)}{4C_h} \in \left[1, \frac{32}{27}\right]$, two δ's satisfy the above first-order condition. However, we can show that one of the δ's that is greater than $\frac{2}{3}$ is a saddle point.

To see this, we check the second-order condition of the hardware firm's problem. First note that $\frac{\partial^2 \pi_h}{\partial p_h \partial f} = \frac{\partial^2 \pi_h}{\partial p_h \partial \delta} = 0$ and $\frac{\partial^2 \pi_h}{\partial^2 p_h} < 0$. Thus, we only need to examine the condition for (f, δ). The Hessian is given by

$$H = \begin{pmatrix} \frac{\partial^2 \pi_h}{\partial^2 f} & \frac{\partial^2 \pi_h}{\partial f \partial \delta} \\ \frac{\partial^2 \pi_h}{\partial \delta \partial f} & \frac{\partial^2 \pi_h}{\partial^2 \delta} \end{pmatrix} = \begin{pmatrix} -\frac{2-\delta}{2(1-\gamma)} & -\frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)} \\ -\frac{\bar{\alpha}(1-\gamma)-f}{2(1-\gamma)} & -C_h \end{pmatrix}.$$

We know that $-\frac{2-\delta}{2(1-\gamma)} < 0$, and

$$\begin{aligned} |H| &= \frac{(2-\delta)C_h}{2(1-\gamma)} - \frac{(\bar{\alpha}(1-\gamma)-f)^2}{4(1-\gamma)^2} \\ &= \frac{2(1-\gamma)(2-\delta)C_h - (\bar{\alpha}(1-\gamma)-f)^2}{4(1-\gamma)^2}. \end{aligned}$$

Since the denominator is positive, we only need to check the sign of the numerator. In order for H to be negative semidefinite, we need

$$\frac{2-\delta}{2\delta} - 1 \geq 0 \Leftrightarrow \delta \leq \frac{2}{3}.$$

Thus, we conclude that when $\frac{\bar{\alpha}^2(1-\gamma)}{4C_h} \in [1, \frac{32}{27})$, one of the solutions for δ that is greater than $\frac{2}{3}$ is a saddle point.

Thus, there is a unique δ that satisfies this condition if $\frac{\bar{\alpha}^2(1-\gamma)}{4C_h} \leq \frac{32}{27}$ (after removing the saddle point), which is equivalent to $\gamma \geq \underline{\gamma} \equiv 1 - \frac{128C_h}{27\bar{\alpha}^2}$. This condition is satisfied under Assumption 2.5. We also showed that $\frac{\partial \delta}{\partial \gamma} < 0$. Thus, we can find a unique $\overline{\gamma}$ such that $\delta(\overline{\gamma}) = \frac{C_s}{2C_h + C_s}$, and condition (3) will be satisfied for any $\gamma \leq \overline{\gamma}$. Substituting $\delta = \frac{C_s}{2C_h + C_s}$ into the above first-order condition, we get

$$\frac{C_s}{2C_h + C_s}\left(2 - \frac{C_s}{2C_h + C_s}\right)^2 = \frac{\bar{\alpha}^2(1-\overline{\gamma})}{4C_h} \Leftrightarrow \overline{\gamma} = 1 - \frac{4(4C_h + C_s)C_h C_s}{\bar{\alpha}^2(2C_h + C_s)^2}.$$

Assumption (2.5) guarantees that $\gamma \leq \overline{\gamma}$.

Finally, since the license fee is positive, it is easy to show that the hardware firm's profits are larger than $\bar{\alpha}v$.

A.4 Proof of Proposition 2.8

Proof. First, we use the Envelope theorem to compute $\frac{\partial \pi_h}{\partial \gamma}$:

$$\begin{aligned} \frac{\partial \pi_h}{\partial \gamma} &= \underbrace{\frac{\gamma^2\bar{\alpha}^2 - v^2}{4\gamma^2}}_{\text{increase in hardware profit}} - \underbrace{\frac{\bar{\alpha}^2}{4(2-\delta)}}_{\text{decrease in software profit}} \\ &= \frac{(\gamma^2\bar{\alpha}^2 - v^2)(2-\delta) - \gamma^2\bar{\alpha}^2}{4\gamma^2(2-\delta)}. \end{aligned}$$

The denominator is positive, so we check the sign of the numerator.

$$\begin{aligned} numerator &= \gamma^2\bar{\alpha}^2(1-\delta) - v^2(2-\delta) \\ &= v^2(1-\delta)\left[\frac{\gamma^2\bar{\alpha}^2}{v^2} - \frac{2-\delta}{1-\delta}\right]. \end{aligned}$$

Thus the sign of the numerator is determined by whether $A(\gamma) \equiv \frac{\gamma^2\bar{\alpha}^2}{v^2}$ is greater/smaller than $B(\gamma) \equiv \frac{2-\delta}{1-\delta}$ (an implicit function of γ through δ). First notice that $\frac{\partial A(\gamma)}{\partial\gamma} > 0$ and $\frac{\partial B(\gamma)}{\partial\gamma} < 0\ \forall\gamma$. The latter can be seen by

$$\frac{\partial B(\gamma)}{\partial\gamma} = \frac{\partial B(\gamma)}{\partial\delta}\frac{\partial\delta}{\partial\gamma}.$$

We know $\frac{\partial B(\gamma)}{\partial\delta} > 0$, and from Proposition 2.6, we have $\frac{\partial\delta}{\partial\gamma} < 0$ at the optimal δ. Then in order for $\frac{\partial\pi_h}{\partial\gamma} > 0$ to happen for some $\gamma \in \Gamma$, we need $A(\gamma) > B(\gamma)$ at the upper bound of Γ, i.e.,

$$A(\overline{\gamma}) > B(\overline{\gamma}),$$

where $\overline{\gamma}$ is defined in Assumption 2.5. Since $\delta = \frac{C_s}{2C_h+C_s}$ at $\gamma = \overline{\gamma}$, we get

$$A(\overline{\gamma}) - B(\overline{\gamma}) = \frac{\overline{\gamma}^2\bar{\alpha}^2}{v^2} - \frac{4C_h+C_s}{2C_h}.$$

Let $C_s = \mu C_h$ for $\mu \in (0,1]$. Then $A(\overline{\gamma}) - B(\overline{\gamma}) > 0$ is equivalent to

$$C_h < \frac{(2+\mu)^2(2\bar{\alpha}^2 - (4+\mu)v^2)}{8(4+\mu)\mu}.$$

In order for this to hold, we require the RHS to be positive and this is equivalent to the conditions: $\bar{\alpha} > \sqrt{2}v$ and $\mu < \frac{2(\bar{\alpha}^2-2v^2)}{v^2}$. The first condition is guaranteed by Assumption 2.1. Note that in the Full Outsourcing case, scenario 1 (interior solution case), this condition also guarantees a non-empty range of γ in which $\frac{\partial\pi_h}{\partial\gamma} > 0$. That is, in that scenario, the increasing profits in γ happens for $\gamma > \frac{\sqrt{2}v}{\bar{\alpha}}$. The second condition shows that the outsourced software development cost needs to be low enough (relative to the in-house software development cost) so that the hardware firm can outsource software production without lowering the license fee revenue (i.e., in order to keep software developer's profits greater than or equal to zero).

Finally, we can show that if $\frac{\sqrt{2}v}{\bar{\alpha}} \in \Gamma$, then

$$A(\frac{\sqrt{2}v}{\bar{\alpha}}) = 2 < B(\frac{\sqrt{2}v}{\bar{\alpha}}),$$

because $B(\gamma) > 2$ for $\gamma \in (0,1)$. In other words, the threshold of γ that changes the sign of $\frac{\partial\pi_h}{\partial\gamma}$ from negative to positive is larger than $\frac{\sqrt{2}v}{\bar{\alpha}}$, the threshold in the Full Outsourcing case. This finding is intuitive because in the endogenous outsourcing case, π_h can be considered as a mixture of π_h in the Integration and Outsourcing cases. Since in the Full Integration case, $\frac{\partial \pi_h}{\partial \gamma} < 0 \; \forall \gamma$, we need a larger γ than that in the Outsourcing case to compensate the large loss that comes from the in-house software profit due to a high γ. □