

Designing free software samples: a game theoretic approach

Christophe Faugère · Giri Kumar Tayi

Published online: 21 February 2007
© Springer Science+Business Media, LLC 2007

Abstract We develop a vertical differentiation game-theoretic model that addresses the issue of designing free software samples (shareware) for attaining follow-on sales. When shareware can be reinstalled, cannibalization of sales of the commercial product may ensue. We analyze the optimal design of free software according to two characteristics: the evaluation period allotted for sampling (potentially renewable) and the proportion of features included in the sample. We introduce a new software classification scheme based on the characteristics of the sample that aid consumer learning. We find that the optimal combination of features and trial time greatly depends on the category of software within the classification scheme. Under alternative learning scenarios, we show that the monopolist may be better off not suppressing potential shareware reinstallation.

Keywords Free sample · Shareware · Optimal software design · Vertical differentiation · Monopolist · Sales cannibalization

1 Introduction

There are several reasons why firms give away free samples of their products. Some firms offer free samples to raise the cost of switching to competing prod-

ucts, others attempt to leverage possible network effects or sell up-grades or complementary products. However, going back to the roots of traditional marketing, a primary purpose of offering free samples is to enhance sales by providing first-hand experience to users [1]. Specifically, a sample allows the consumer to learn about the product.¹ When the learning experience is positive it usually results in increased sales.

In this paper, we analyze the case of a company that sells information products and gives away free software samples in order to build product awareness and attain follow-on sales. These information products include mass consumer software and typically are single-user applications deployed on non-networked devices. Some examples include computer games, music/photo/art software, spreadsheets, word-processors, and web-design and business application software. For many of these products, distributing shareware is the most widely used marketing method [3]. Moreover, the value of these products' *intrinsic* features becomes more evident via using the sample.

By contrast, free samples such as Adobe Acrobat Reader and MS PowerPoint viewer are designed mainly to take advantage of the network effects stemming from a large installed customer base [9]. These network effects can be regarded as *extrinsic* features of the product. Our focus is on the intrinsic features of software products for which network effects are not necessarily the dominant factor.

C. Faugère · G. K. Tayi (✉)
School of Business, University at Albany, SUNY, Albany,
NY 12222, USA
e-mail: g.tayi@albany.edu

C. Faugère
e-mail: c.faugere@albany.edu

¹ Learning can be broadly defined here as acquiring information about previously unknown characteristics of the product, or learning how to use the product in an effective manner (i.e. training).

The nature of traditional physical samples differs from that of information products such as software. Typically, a physical sample provides a limited experience of the actual product. Even in the case of a durable good like an automobile, a sample in the form of a test drive gives an experience of the majority of features that the customer needs, but only for a limited time. However, in the case of shareware, the consumer can often reinstall the sample on a repeated basis, especially when the sample offers the product's most useful features. In that instance, the distribution of free samples leads to sales cannibalization.²

Consequently, the sample evaluation period, which controls the frequency of reinstallation and the proportion of features included in the sample are critical characteristics for the design of free software.³ RealOne player is an example where the software's free basic version comes with limited functionality and the company periodically posts intrusive messages on the computer desktop urging the consumer to purchase the unrestricted version. Other pieces of software like Adobe Photoshop run out after 30 days of free trial. Although many shareware come with these restrictions, an adept user can in many instances, reinstall the free version anew by uninstalling the previous version and setting the counter back to zero. This activity is generally considered a form of piracy.⁴

Here, we examine how a monopolist optimally designs free software samples. We address the issue of sales cannibalization while focusing on the intrinsic features of the software product. Consumers learn about the product and may then decide to purchase the product or continue to use the sample by circumventing the evaluation period restriction. An obvious

application of our model is to B2C or B2B mass consumer markets that offer shareware. However, our model also covers specialized markets such as industrial software.

We introduce a new software classification scheme based on shareware characteristics, which helps consumers assess the novelty and performance of the product as well as other key properties such as ease of use. For each category of software, we identify the optimal combination of trial time and proportion of features to be included in the free samples. Our results are confirmed by actual marketing practices of software vendors. Marketing managers can also use this classification scheme as a normative guide for the design of new shareware.

Furthermore, we show that when consumers' learning happens in stages, it may be beneficial for the monopolist *not to deter* reinstallation. The reason is that by keeping the initial trial time short, revenues can be extracted faster from the segment of consumers that are willing to buy at the early stage, while generating additional revenues from the consumers who take longer to develop an appreciation for the product.

The remainder of the paper is organized as follows: Section 2 introduces the game-theoretic model. Section 3 introduces a new software classification scheme based on the key parameters of the model. Section 4 presents the main propositions describing the game's equilibria under the base case scenario about learning. Section 5 extends the model to multi-period learning. Our concluding comments appear in Section 6.

2 The basic model

We use a vertical differentiation game-theoretic model [14]. In a vertical differentiation framework, consumers differ according to their reservation prices but have a unanimous preference ranking over the product's quality/attributes. Hence, the producer's objective is not only to determine the optimal price, but also to select the optimal quality/attributes of his product as well, since these attributes drive demand.⁵

This framework is well suited for studying the decision of offering free software samples alongside a commercial product. In practice, choosing the attributes of a software sample is a non-trivial task, since (1) software samples are in many instances durable goods, and (2) sample attributes may cannibalize the sales of the commercial product when too many of the

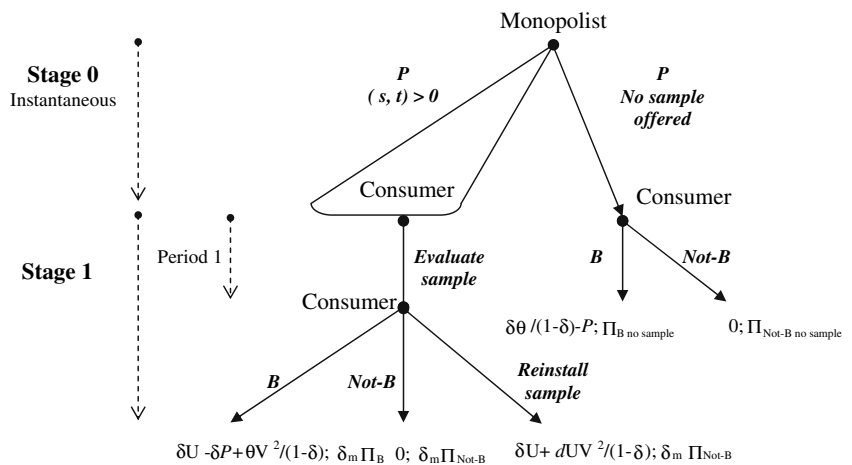
² The issue of product sales cannibalization is also studied in Takeyama [13]. She shows that a durable-good monopolist may indeed choose to cannibalize its own high-end product by offering goods of lesser quality, which results in a consumer welfare improving equilibrium. Haruvy and Prasad [6] analyze sales cannibalization in the context of network effects. They find that when software products benefit from network effects, these effects may play a role in mitigating the cannibalization of sales. However, they do not consider the case of network effects having an adverse impact on sales. In particular, a large base of free software users may create an incentive for newcomers to keep using the free sample rather than buying the actual product.

³ Cakanyildirim and Dalgic [2] study demonstration strategies that promote information products. The strategic variables used in their model are demo features and the length of the demo phase. Heiman and Muller [8] show how the demonstration phase may vary in length depending on how competitive the industry is, and depending on whether the demonstration is personalized or not.

⁴ Software & Information Industry Association's website at <http://www.siiia.net/piracy..>

⁵ For a good introduction on vertical product differentiation, see Sutton [12].

Fig. 1 The basic game in extensive form



product’s features are included in the sample.⁶ We assert that the proportion of features included in the sample and the evaluation period together help the consumer learn about the characteristics of the actual product. Hence, modeling the consumer’s learning process plays a key role in our analysis.

The software industry is generally concentrated. For example in 2005, the *top two* companies in the desktop application industry represented about 82% of the industry’s revenues.⁷ This is because the up-front cost of developing software products is prohibitively high. Presumably, product design is mainly achieved via R&D-related expenditures (fixed costs) with little to no increase in marginal cost. We adopt a monopolistic market structure for our model, since our primary goal is to analyze the optimal design of free samples within the simplest framework that captures market concentration.

The basic model consists of a two-stage sequential game with imperfect information about the product. The game tree is described in Fig. 1 and all the notations used in the basic model are summarized in Table 1 below. There are two categories of players: a monopolist and a population of consumers. Figure 1 depicts the game tree and the payoffs for a consumer and the monopolist. A standard game period, normalized to one, is interpreted as the industry standard for the *maximum* duration of time a consumer is allowed to use a free sample.⁸ In stage 0, the monopolist chooses the commercial product’s price P , and decides

whether to offer a free sample or not. This stage is instantaneous. Next, Stage 1 has an infinite number of periods. Introducing an infinite number of periods is a useful device to account for the durable nature of digital goods. The consumer’s strategic move occurs either at the beginning or at the end of period 1, as shown in Fig. 1. During the remaining periods (2 to ∞), the consumer is passive and draws utility following her strategic move in period 1.

At the beginning of Stage 1 (and period 1), a consumer may decide to buy (B) or not to buy (*Not-B*) the product, when the monopolist has *not* offered a free sample. On the other hand, when a free sample is offered, a consumer evaluates the sample in period 1 and then makes a decision. In that case, we refer to period 1 as the *learning* period. The rationale for modeling consumer behavior that way, is that sampling before buying is a widespread practice regarding information goods, since free samples are easily obtainable and evaluation periods are relatively short. Moreover, vendors tend to distribute free samples to the segment of consumers who are expected *not to buy* the product right away.

At the end of period 1, a consumer faces the decision to buy (B) or not to buy (*Not-B*) the commercial product. A third option for the consumer is to not buy the product and continue using the sample as a substitute for the product itself (*reinstall the sample*).⁹ From period 2 on, she receives an infinite stream of

⁶ Coase [4] and Stokey [11] are the classic articles regarding the analysis of the standard durable good monopolist. More recently, Dhebar [5] studies how the speed of improvement of the intrinsic quality of software (or other durable good products) affects the dynamics of sales.

⁷ Source: “Scale or Scope: that is the M&A Question in Software”, Deloitte Consulting, 2005.

⁸ In extensive form game terminology, a stage is a collection of periods defined by which player’s turn it is to play strategically.

⁹ To simplify the analysis we assume that while the consumer tests the sample during the evaluation period, which could be less than the full game period; she makes his decision to buy (or not to buy) the product or re-use the sample only at the end of period 1. In Sect. 5, we examine other types of learning processes, which may result in different segments of consumers buying in different periods.

Table 1 Notations for the basic model

Symbol	Description
N	Size of the market (population size)
θ	Prior reservation price/preference parameter. $\theta \in [\theta_L; \theta_H]$.
$t_0; 1$	Minimum and maximum evaluation periods within one game period
$s_0; 1$	Minimum and maximum proportions of features made available in sample
d	Utility reduction parameter due to sample reinstallation. $d \in [0; 1]$
A	Maximum learning effect on the reservation price after period 1. $A > 0$
α	Marginal effect on learning of additional features included in the sample
δ	Consumer's discount rate $\in [0; 1]$
δ_m	Monopolist's discount rate $\in [0; 1]$
c	Marginal cost of producing one copy of the <i>commercial</i> software product
γ	Fraction of the cost of distributing of copies of the full commercial product's to measure the sample's distribution cost. $\gamma \in [0; 1]$
F	Fixed cost of producing commercial product and sample
P	Price of the commercial software product
s, t	Free sample attributes: respectively the proportion of features and trial time
U	Direct period-wise utility from using the sample
V	Learning factor that impacts the reservation price θ . It is a function of s and t
Π_B	Monopolist's profit when consumer samples and then buys the product

utility payoffs either from the product if she buys or from reinstalling the sample. Not buying brings zero utility. Payoffs are present values of future expected utilities discounted as of the beginning of Stage 1.

2.1 The consumer

A consumer's utility depends on the attributes of the software she is using. A free sample is a version of the commercial product that is scaled down along two dimensions: the proportion of features s available in the free sample and the evaluation period (or trial time) t . Both variables s and t take values in the unitary interval $[0, 1]$. For example, the trial version of Scientific Word is crippled (saving function unavailable) compared to the full-blown version, and the evaluation period is 30 days.

It is important to emphasize that the sample serves a dual purpose for the consumer. First, as a scaled down version of the commercial product it provides *direct* utility from straight use. The second function of the sample is to enable the consumer to learn about the scope and performance of the product. A simple analogy is that a digital movie clip serves as a sample for the full-length movie DVD. Consumers may enjoy watching the clip for its own sake, but it also conveys information about what is to be expected in the full-length feature.

Learning can take several forms: consumers may use the sample as a training device to operate the product and they may as well discover features of the software, which are novel to them. In our framework, learning is modeled as an updating of the consumers' prior valuation of the product, based on receiving new information from the sample.

Formally, we denote the *direct* one-period utility from using a sample as $U(s, t) = \theta s((1 + t)/2)$ for $s > 0$ and $t > 0$; $U = 0$ otherwise. Each consumer has a preference parameter $\theta \in [\theta_L; \theta_H]$ that represents her *prior* reservation price for the product in the absence of learning. This parameter is uniformly distributed according to $\Psi(\theta) = (\theta - \theta_L)/(\theta_H - \theta_L)$. The utility function U depends on the proportion of features s included in the sample and the trial time (evaluation period) t .¹⁰ Our formulation captures the reasonable property that consumers derive "more utility" from the proportion of features than trial time. Specifically, in instances where the evaluation period may be short, consumers still derive a high level of utility when the sample includes a large proportion of features. By contrast, consumers generally derive little utility from a sample having a small proportion of features, even though a long evaluation period may be given. While using the sample in period 1, consumers learn about the product, its performance and its intrinsic quality. We assume that learning raises the reservation price θ by a multiplicative variable V , which is referred to as the learning function:¹¹

¹⁰ Note that the *one-period* utility obtained from using the full commercial version is θ since the commercial product provides all the features ($s = 1$) and gives the full trial time ($t = 1$). However, a consumer may still face a time constraint, even though full features and full trial time are being given to him. This time constraint arises from the effort the consumer must spend in reinstalling the sample, as we will see later.

¹¹ Our characterization of learning is congruent with Heiman and Muller's [7] approach. They argue that demonstrations (length or trial time) reduce the purchasing risk and thus possibly increase the probability of purchase at any given price. In our model, the likelihood of purchase is impacted not only by trial time but also by the sample's features.

$$V = \begin{cases} 1, & \text{when } s < s_0 \text{ or } t < t_0 \\ A(\alpha s + (1 - \alpha)t) + 1, & \text{when } s_0 \leq s \leq 1 \\ & \text{and } t_0 \leq t \leq 1 \end{cases}$$

The function V depends on the proportion of features and trial time as well. As long as s and t are below their respective minima s_0 and t_0 , the sample is deemed uninformative and the prior reservation price for the commercial product does not change. However, when s and t are above their respective minima, learning occurs and the updated reservation price rises. The parameter A is assumed positive. The parameter $\alpha \in [0, 1]$ represents the impact that the proportion of features has on the learning process relative to trial time. The function V is essentially a mechanism that updates the prior reservation price θ based on new information received from trying the free sample.

Let us now describe the consumers' payoffs. As Fig. 1 indicates, consumers make their first possible strategic move in Stage 1, at the beginning of period 1. At that point, they know their prior reservation price θ for the product. Thus, the one-period *expected* utility from purchasing the product is θ , since no new information about the product is acquired. In that case a consumer's dynamic payoff from buying the product right away is the present value of expected utilities from the product minus the price paid $\delta\theta/(1 - \delta) - P$ if she buys, and 0 otherwise; where the parameter $0 \leq \delta < 1$ denotes the consumers discount rate.¹²

On the other hand, when the monopolist offers a free sample, consumers use the free sample and learn about the commercial product and hence their prior reservation price θ will be updated at the end of period 1. Therefore, after using the sample with a given trial time and proportion of features, the total payoff for a consumer is: $\delta U - \delta P + \theta V \delta^2 / (1 - \delta)$. This payoff comprises of the discounted direct utility δU of using the free sample in period 1; minus the discounted price of the product δP ; plus the expected present value of future utilities from owning the commercial product $\theta V \delta^2 / (1 - \delta)$; given that learning has raised the reservation price to θV in each subsequent periods of Stage 1.¹³

¹² Recall that the utility for the commercial product θ is obtained at the end of each game period.

¹³ In the unlikely instance where the monopolist would give out the full-blown version of the product as the sample, we observe that the utility from buying the product outright is less than that of evaluating the sample and then buying. The main reason for that difference is that expected payoffs are updated based on new information in the latter case. On the other hand, if there were perfect information about the product, no updating would take place and the two cases would yield the same payoff.

Once a consumer has evaluated the sample, he may choose not to buy the product and to reinstall the sample for further use. Her total expected payoff becomes: $\delta U + dUV\delta^2 / (1 - \delta)$. In this case, the payoff is comprised of the discounted direct utility from trying the sample δU during period 1; plus the present value of expected utilities from reinstalling the sample $dUV\delta^2 / (1 - \delta)$ in periods thereafter.¹⁴

The parameter $0 \leq d \leq 1$ represents a scaling factor that reduces the utility of re-using the sample. The main reason for including this parameter is that in general a software developer will take steps to minimize the unfettered reinstallation of his free sample. Thus, consumers must exert some effort to reinstall the sample in any given period.¹⁵ A smaller value for d means a greater amount of effort and hence a greater reduction in the consumer's utility. The parameter d is treated as exogenous here, although in practice a monopolist may optimally choose to make the reinstallation procedure hard. Preventing reinstallation can take several forms. Some common forms involve including anti-piracy code in the sample and/or obtaining ID keys online from the monopolist. In the case where the commercial software product is highly technical and specialized, preventing reinstallation is also achievable via human monitoring.

2.2 The monopolist

Similar to the vertical differentiation model presented in Tirole [14], the monopolist faces a demand $D(P, s, t) = N[1 - \Psi(\theta(P, s, t))]$, where N is the size of the total customer market and P is the product price. The demand is constituted by the segment of consumers whose preference parameter θ is greater than a threshold $\theta(P, s, t)$ determined by the strategic response of consumers to the monopolist's decisions.

We posit that the monopolist is developing the sample in conjunction with the commercial version of the software product. The monopolist's profit is given by $\Pi = D(P, s, t) \times [P - c] - F - N\gamma c$. The parameter c

¹⁴ Note that the learning function V is assumed to also affect the utility obtained from reinstalling the sample, since the consumer may develop a greater appreciation for the sample itself, once the learning period is completed.

¹⁵ It is important to note that in our framework when the monopolist gives the full commercial product as a free sample ($s = 1, t = 1$) with no reinstallation restriction ($d = 1$), demand collapses and the price is zero. However, in the case where $d = 1$, we recognize that trial time is actually of no consequence for the consumer. This boundary case can be easily accommodated by modifying the sample's definition to state that the minimum trial time t_0 is always 1 when $d = 1$.

represents the marginal cost of producing copies of the commercial product. The parameter F is the fixed cost tied to research and development and the design of both the product and sample. This fixed cost is assumed independent of the particular proportion of features or amount of trial time included in the sample.

The other fixed cost $N\gamma c$ is associated with producing and distributing copies of the free sample to the entire population; where $0 \leq \gamma \leq 1$ is a scaling parameter. For example, the monopolist could mail free CDs to the population (e.g. AOL). This latter cost represents a fraction of what the cost of producing and distributing commercial copies to the whole population would be. Obviously, this cost is zero if no samples are made and distributed. It can also be negligible if the marginal cost c is near zero, for example, when shareware is distributed electronically over the internet. Lastly, when the monopolist offers a free sample, his profit is postponed to the end of period 1 and thus his profit is discounted at the rate $0 \leq \delta_m < 1$. A large value of δ_m implies that the monopolist is patient, and vice versa.

3 A new software classification scheme

In this section, we introduce a new software classification scheme based on the characteristics of the shareware.¹⁶ Why focus on the shareware characteristics and not directly on those of the actual software product? The reason is that trial time and proportion of features are choice variables for the monopolist in a sample. These variables give a software vendor the latitude to demonstrate the properties of the software and thus provide surrogate measures for key characteristics of the commercial product.

For example, consumers can assess whether the product is simple or complex to operate, based on how much minimum time is needed to reach basic proficiency in operating the product. This property is naturally reflected in the sample design. Consumers can also evaluate whether functionalities are mostly integrated or can be operated in independent modules. This second property is reflected in the minimal proportion of features included in the sample to facilitate learning. Moreover, the degree of novelty inherent in

the software is reflected by the impact that the sample has on the consumers' reservation prices.¹⁷

Our classification scheme is based on the parameters of the learning function V introduced earlier.¹⁸ We first differentiate between *basic* and *complex* software. Software is qualified as basic when the minimum evaluation period t_0 needed to preview the product is small; and vice versa for a complex product. Software is referred to as *modular* when the minimum proportion of features s_0 necessary for learning about the software is small. For example, free samples of a multi-level game do not usually include all the levels of play. On the other hand, software is defined as *integrated* when the sample requires most of the product's features be included (s_0 large), for example as in the case of a CAD program.

We define a *conventional* software product as one where sample use does not result in significant learning (A small); that is, posterior expectations about the value of the product do not rise dramatically after sample use, holding other parameters constant. On the other hand, we define *novel* software as one where sample use significantly enhances the reservation price (A large), by demonstrating the product's novel features and valuable attributes. For example, web-based business software may contain several features whose novelty can only be appreciated via sample use.

Lastly, we recognize that *individual* features of software products may impact consumers' perceived value with various levels of intensity. We differentiate between software products that are characterized by *high-intensity* features (α high) versus *low-intensity* features (α low). High-intensity software products contain individual features that have high marginal contribution and are critical to achieving the purpose of the software. For example, antivirus or antispyware software would belong in this category, since each feature (scan, quarantine, auto-protect, etc...) contributes significantly to the overall purpose of the software.

By contrast, several features embedded in word processors (strikethrough, font color, etc...) could be viewed as low-intensity features, since such features, although essential, are not critical for the core purpose of the product. Table 2 below presents the classification

¹⁶ Parker and Van Alstyne [10] also introduce a classification scheme for information product design. They classify information goods as strategic complements or substitutes according to their ability to generate network externalities.

¹⁷ Furthermore, all these characteristics are of interest to the consumer mainly prior to the purchase decision. Once the product is purchased it is too late so to speak, since consumers must now contend with the difficulties of learning how to operate the product. Tests featured in consumer magazines are another way to reveal some of the products characteristics to prospective buyers.

¹⁸ This classification is not exclusively tied to the specific functional form we used to model learning, since the parameters have very natural and general interpretations.

Table 2 Software classification scheme

	Modular (s_0 low)		Integrated (s_0 high)	
	Low-intensity (α low)	High-intensity (α high)	Low-intensity (α low)	High-intensity (α high)
Conventional (A low)				
Basic (t_0 low)	Drawing/painting Basic calculator	Music creation Educational	Word processor Basic photo-editing	CAD Utilities software
Complex (t_0 high)	Spreadsheet Personal money	Statistics/optimization Engineering software	Business accounting Data mining	Tax software SAP
Novel (A high)				
Basic (t_0 low)	Multilevel games	Educational	Photo editing Website design	Antivirus/spyware Utilities software
Complex (t_0 high)	Web-fulfillment	Stock trading Engineering software	Supply chain mngt.	Plant software

scheme by partitioning the parameter space into its finest granularity and gives some illustrative examples.

4 Equilibrium analysis of the basic model

This section presents the Subgame Perfect Equilibria of the basic game. The following propositions describe the monopolist’s optimal decision whether to offer a free sample or not. When a free sample is offered, we analyze the optimal design of the free sample according to our software classification scheme given in Table 2. We focus on pure strategy equilibria, which involve a segment of consumers buying the product either right away if no free sample is given, or buying the product after trying the sample if one is given. However, it is important to first show that *no* Subgame Perfect equilibrium exists where consumers would commit to buy the product right away even when a free sample is being offered.

Proposition 1 (Sampling before buying) *Assume that consumers commit to buy the product right away at the beginning of period 1 (Stage 1), then the monopolist should not offer any free sample.*

Proof If consumers commit to buying the product right away, they must be indifferent between receiving a free sample or not. Thus, the monopolist should not offer a free sample, since in doing so his profit will be larger as he will not incur the manufacturing and distribution costs of the sample. □

Thus, the only possible equilibrium where consumers buy the product right away is one where the monopolist does not offer a free sample. The next proposition describes the optimal monopoly profit and prices in the cases where the consumer buys the

product immediately when no free sample is offered, or buys the product after sampling.

Proposition 2 (Monopoly pricing and profit) *Assume that the demand is non-empty $\theta_H > c(1 - \delta)/\delta$, then the optimal monopoly price and profit are: $P_1^* = (c/2) + (\delta K^* \theta_H / (2(1 - \delta)))$ and $\Pi_1^* = (N / (\theta_H - \theta_L)) \times (\delta K^* / (4(1 - \delta))) \times [\theta_H - (((1 - \delta)c) / \delta K^*)]^2 - F - N\gamma c$; with $K^* = K(s^*, t^*) = [1 - ds^*((1 + t^*)/2)]V^*$ and $K^* > 1$ when a free sample is offered; otherwise when no free sample is offered, substitute $K^* = 1$ and $N\gamma c = 0$ in these expressions.*

Proof See Appendix A, general Cases 1 and 2.

Proposition 2 shows that the optimal price and profit are functions of standard economic variables such as the consumers’ discount rate, the marginal cost of producing copies of the product and the highest and lowest reservation prices in the population. Note that the monopoly profit and price are also increasing functions of the variable K^* , which plays a key role in the analysis. In the case where a free sample is offered, K^* represents the incremental utility effect generated by the sample, which pushes consumers to buy the commercial product. It is a function of the optimal proportion of features s^* and trial time t^* . On the other hand, when no free sample is distributed, the value of $K^* = 1$.

For example, take the case of an innovative product. Offering a free sample may simultaneously enhance sales as well as cannibalization of sales. The value of K^* satisfies $K^* > 1$ when the former effect dominates. The monopolist charges a higher price because he anticipates that the product becomes more attractive since the consumer tries the sample. This, in turn lowers quantity demanded, while at the same time, demand expands (condition $\theta > (1 - \delta)P/\delta K(s, t)$ in

Appendix A). The reason is that learning raises the reservation prices above the buying threshold for a segment of consumers, who otherwise would not have bought the product. Thus, even though there is greater incentive to reinstall the sample of an innovative product, sales expansion still wins over cannibalization when $K^* > 1$.

Proposition 3 (No free sample is offered) *Assume the monopolist is impatient (δ_m small), that parameters A and α are small, s_0 and/or t_0 is close to 1 and the reinstallation cost is low (d large). It is always optimal for the monopolist not to offer a free sample.*

Proof See Appendix A, Case 2.

This proposition illustrates the boundary case where it is not advantageous for the monopolist to offer a free sample. Although the monopolist can sell the product, offering a sample would be detrimental. The conditions that lead to this outcome are that the sample would not be novel enough, that the minimal proportion of features has to be large, and reinstallation would be too easy. These conditions, contribute to either making learning inconsequential, or to drive all consumers toward reinstalling the sample, resulting in a loss for the monopolist.

The next three propositions analyze the optimal mix of proportion of features and trial time which are to be included in a free sample. We assume that the monopolist is patient enough (δ_m large enough), to rule out the trivial case where no sample would be offered over the entire parameter space. These propositions are formulated based on the software classification scheme developed in the previous section. These propositions provide direct managerial insights into the design of each category of software samples.¹⁹

Proposition 4 (Non-durable free sample) *Irrespective of the parameter values for A , α , s_0 and t_0 , it is always optimal for the monopolist to offer a free sample with full features, when the reinstallation cost is infinite (d near zero).*

Proof See Appendix A, Case 1.1.

This proposition deals with the simplest case of non-durable software samples. When the monopolist can fully prevent reinstallation of free software samples, it

¹⁹ It is also important to note that as shown in Appendix A (Case 1), the local optima (trial time and proportion of features) are independent of both the consumer's and the monopolist's discount rates, since in our framework profit maximization amounts to maximizing static consumer demand. However, the monopolist's discount rate does play a role in determining the global optimum (conclusion of Case 2).

is trivial to show that the sample should include 100% of the features present in the commercial version. We will examine later the types of software for which non-durable free samples are offered. In these cases, human monitoring arises as a natural means of preventing unauthorized reinstallation of free samples. These software products are typically sold in small and specialized markets with direct and personalized vendor–customer relationships.

Proposition 5 *In the case of conventional (A small) software, optimal free sample design is characterized by the following:*

- (i) *For low-intensity modular software, a free sample should have $s^* = s_0$ and $t^* = 1$, no matter how easy it is to reinstall the sample.*
- (ii) *No free sample should be offered for low-intensity integrated software.*
- (iii) *For high-intensity software, a free sample should have $t^* = t_0$ only when the reinstallation cost is significant (d small). Otherwise, no free sample should be given.*
- (iv) (a) *For high-intensity modular software, a free sample should have $s^* = 1$ only when d is small, otherwise $s^* = s_0$; whereas (b) a high-intensity integrated software should have a sample with $s^* = s_0$.*

Proof See Appendix A: (i) corresponds to Case 1.4; (ii) corresponds to Case 2; (iii) corresponds to Cases 1.2 and 1.3; (iv) (a) is Cases 1.3 and 1.4 and (b) is Case 1.2. \square

Proposition 5 analyzes the design of free samples for various types of software according to our classification scheme, with a special focus on conventional software. Interestingly, this proposition is supported by actual marketing practices of several software products. For example, case (i) applies to spreadsheet and drawing/painting software and states that for these products a free sample should include the minimum proportion of features and the maximum evaluation period. Real world examples are the Paintshop Pro evaluation copy and MS-Works. MS-Works is offered as part of Windows based PCs' basic software package. It has no time restriction but offers limited features as compared to MS-Excel.

Case (ii) is relevant to word processors, standard business accounting or database software, and states that no free samples should be offered for these products, as observed from industry practice.²⁰ As can

²⁰ Obviously there are other circumstances when free samples should not be given. Case 2 in Appendix A shows that this could happen if the cost of distribution is significant and the monopolist is impatient.

be seen from cases (iii) and (iv) the proportion of features plays a prominent role in the design of the free samples for high intensity software. While trial time is capped at the minimum, the proportion of features varies for each sub-category of high-intensity software. For instance, case (iii) holds for music creation, utilities and engineering software. Case (iv) (a) holds for statistics/optimization software, (b) CAD or tax software. In the case of CAD software, it appears that giving a 30-day trial period is an industry norm. A 30-day trial period is at the low end of the spectrum of standard evaluation periods.

Proposition 6 *In the case of novel (A large) software, optimal free sample design is characterized by the following:*

- (i) *No free sample should be given in the case of integrated complex software, unless the sample is hard to reinstall (d close to 0).*
- (ii) *Low-intensity software should offer free samples with $s^* = s_0$, except for case (i).*
- (iii) (a) *For low-intensity modular software, a free sample should have $t^* = 1$, while (b) low-intensity integrated software samples should have $t^* = 1$ when d is small, and $t_0 < t^* < 1$ otherwise.*
- (iv) *For high-intensity software, a free sample should have $t^* = t_0$, no matter the value for d , and should have $s^* = 1$ when d is small, and $s_0 < s^* < 1$ otherwise.*
- (v) *For high-intensity software, everything being equal, basic products should have samples with larger values of s^* as compared to samples of complex products.*

Proof See Appendix A: (i) corresponds to Case 2; (ii) corresponds to Case 1.2, 1.4 and 1.6 combined; (iii) (a) corresponds to Case 1.4 and (b) to Cases 1.4 and 1.6 combined; (iv) is Case 1.2, 1.3 and 1.5 combined; (v) follows from comparative statics on Eq. 7 in Appendix A. \square

Proposition 6 offers insights into the optimal design of free samples for novel software. Case (i) applies to B2B markets such as industrial software, for example supply chain management or plant software products. There is good evidence to indicate that these integrated and complex software products have highly technical and specialized niche markets, compared to other products. In these markets, the issue of unauthorized reinstallation may not be as chronic, since a vendor will distribute free samples to a closely monitored clientele. This monitoring reduces the likelihood of breaking evaluation period agreements. Thus by

nature, we should expect the parameter d to be close to zero, in these markets. Hence, in conjunction with Proposition 4, these software types should also offer 100% of the products features.

Previously in Proposition 5, we found that *no* free samples were given for low intensity integrated software. On the other hand, in Proposition 6 cases (ii) and (iii) show that free samples should be offered for this type of software unless it is complex and/or easy to reinstall. Since these products are novel, they benefit from a greater ‘wow’ effect produced by the sample. It is then crucial to restrict the other dimensions that would make reinstallation attractive. This is done by offering the minimum proportion of features, and further shrinking the evaluation period the more integrated the product is. Case (ii) is relevant for multi-level games and photo editing software. Case (iii) (a) holds for web fulfillment software which should offer the maximum trial time versus (b) that applies to website design software, where trial time may be more limited. In the case of web fulfillment shareware, it appears that a common practice is to restrict the number of uses or launches. Interestingly, this strategy amounts to giving a large trial time since users may try the shareware over several months.

In case (iv), and contrary to Proposition 5, we see that offering free samples for high-intensity products occurs as a rule. This case holds for educational and antivirus software, which should offer the minimum trial time and a smaller proportion of features the easier it is to reinstall the sample. Case (v) shows for example that a basic software sample geared to stock market education should offer a larger proportion of features as compared to a stock-trading software sample, which is more complex.

Interestingly, for novel software the proposition draws a sharp contrast between low-intensity versus high-intensity software. Cases (ii–iv) reveal that trial time is the key strategic variable for low-intensity software, whereas proportion of features is the key variable for high intensity software. The chief reason is simply that the impact of varying the proportion of features is small for low-intensity vis-a-vis high-intensity software samples. Overall, Propositions 5 and 6 not only confirm current marketing practices for various types of software; they also offer normative advice for any new software product that may fall into one of the categories of our classification scheme.

5 Multi-period learning and sample reinstallation

Software companies go to great lengths to deter or at least monitor free sample reinstallation. Some pro-

ducers require online registration and can, to some extent, control the re-use of their free sample. In this section, we show that when consumers' learning occurs over multiple periods, extending the initial evaluation period while completely preventing reinstallation thereafter may actually harm the monopolist.²¹

In general, it is clear that giving a short initial evaluation period is good business practice since everything else being equal; a software company would rather generate revenues sooner than later. The initial trial time is a useful device to coax the consumer into making a decision whether to buy or not. Still, some consumers may feel that they did not have enough time to fully learn about the product at the end of the initial evaluation period. In that instance, it would appear that extending the length of the evaluation period is a natural way of resolving this issue. However, giving a longer initial trial time may not be the right action to take, since in doing so the monopolist may *delay* sales revenues from consumers who were willing to buy the product at an earlier point.²²

In this section, we model a simple version of this issue. For tractability of the analysis, we study the impact of changing the *maximum* trial period on the monopolist's profit.²³ We still focus on the case where reinstallation is exogenous and costly to consumers. However, we now posit that incremental learning may take place over two unitary periods instead of a single period. This enables us to show that there are particular types of software for which a monopolist gains more by *not* completely deterring shareware reinstallation, rather than extending the initial trial period and preclude reinstallation.

The basic model is modified as follows. The game period (or maximum trial period) is now denoted by $T > 0$. We consider two possible scenarios. In the first scenario, the game period is of unitary length as before, i.e. $T = 1$. However, the consumer's learning accretes

²¹ The concept of multi-period learning may seem misleading at first, since a 'period' can be arbitrarily defined to fit any length necessary. However, the choice of the industry standard regarding the maximum evaluation period must be determined in part by how long it takes consumers on average to learn about the product. In addition, a monopolist may choose to extend the evaluation period to expand demand, when for example there is a wide dispersion of learning abilities in the consumer population.

²² However, this loss may itself be mitigated, when for example, the monopolist is able to prevent further reinstallation of the sample.

²³ We still assume here that the consumer will only *buy* at the end of a game period (maximum industry standard), rather than at the end of trial time. In that instance, the monopolist profits are always occurring at the end of the game period. Thus, a delay in collecting revenues is possible, but it only depends on the *maximum* trial duration.

over *two* periods and is represented by a new learning function L_1 :

$$L_1 = \begin{cases} 1, & \text{when } s < s_0 \text{ or } t < t_0 \\ V, & \text{when } s_0 \leq s \leq 1 \text{ and } t_0 \leq t \leq 1 \\ & \text{over period 1.} \\ \lambda V, & \text{when } s_0 \leq s \leq 1 \text{ and } t_0 \leq t \leq 1 \\ & \text{for all subsequent periods,} \end{cases}$$

The function L_1 is a modified version of the learning function V that was introduced in the basic model. The parameter $\lambda > 1$ represents the effect of accretive learning. A justification for that effect is that consumers can now extract more information after having familiarized themselves with the sample in the first period. Essentially, it is a form of learning-by-doing. We assume that incremental learning stops beyond period 2, so that a consumer's updated valuation of the product remains capped at the same level in subsequent periods.

In the second scenario, we assume that the maximum trial time is extended to $T \in [1, 2]$, and the monopolist only gives *one* period of length T for consumers to learn about the product before buying. In that case, the learning function becomes $L_2 = \lambda^{T-1}V$ for one-period. We also assume that the sample *cannot* be reinstalled after that, so that the parameter $d = 0$.²⁴ The basic idea is to contrast two scenarios: (1) where multiple reinstallations of the sample are possible with greater learning as the outcome and (2) where the maximum evaluation period may be expanded but reinstallation is impossible. Figure 2 illustrates the two time lines corresponding to each scenario.

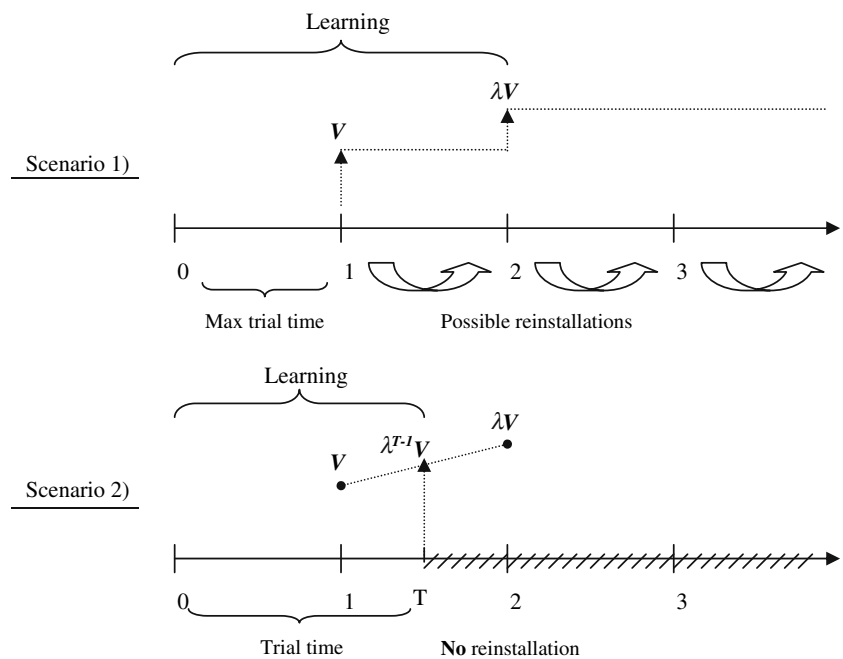
Proposition 7 (Monopoly pricing and profit) *Assume that the demand is non-empty $\theta_H > c(1 - \delta)/\delta$; then the optimal monopoly price and profit are: $P^* = (c/2) + (\delta K^* \theta_H / 2(1 - \delta)) \times (\lambda / (\lambda - \delta_m(\lambda - 1)))$; and $\Pi^* = (N / (\theta_H - \theta_L)) \times (\delta K^* / 4(1 - \delta)) \times (\lambda / (\lambda - \delta_m(\lambda - 1))) \times [\theta_H - ((\lambda - \delta_m(\lambda - 1)) / \lambda) \times (((1 - \delta)c) / \delta K^*)]^2 - F - N\gamma c$, in the case of scenario 1 where reinstallation is allowed. In the case of extended trial time without reinstallation (scenario 2), the optimal monopoly price and profit are: $P_3^* = (c/2) + (\lambda^{T-1} \delta^T V_{\max} \theta_H / 2(1 - \delta^T))$ and $\Pi_3^* = (N / (\theta_H - \theta_L)) \times (\lambda^{T-1} \delta^T V_{\max} / 4(1 - \delta^T)) \times [\theta_H - ((1 - \delta^T)c / \lambda^{T-1} \delta^T V_{\max})]^2 F - N\gamma c$.*

Proof See Appendix B, general Cases 1 and 2.

Note that with multi-period learning, in both scenarios 1 and 2, the price the monopolist is charging

²⁴ Implicitly, we are assuming that a monopolist can prevent 100% reinstallation at a zero fixed cost, via human monitoring for example.

Fig. 2 Time lines for extended learning scenarios



and its profit are both increasing functions of the learning parameter λ . In other words, the more effective prolonged learning is, the more valuable the product becomes for the consumer, since she gets the most information out of the trial time. The next proposition compares the two scenarios and gives parameter ranges for which one scenario may dominate the other from the monopolist’s standpoint.

Proposition 8 For any feasible combination of parameters $(A, d, \alpha, t_0, \delta, \delta_m, \lambda)$, there exists a corresponding threshold trial period $\bar{T} \in [1, 2]$, such that:

- (i) Let incremental learning be small (λ close to 1), and the sample is easy to reinstall (d large). The monopolist is better off giving a free sample with $s_0 \leq s^* \leq 1$ and $t_0 \leq t^* \leq 1$ and possible reinstallations of unitary length, rather than precluding reinstallation and extending the maximum trial period beyond \bar{T} (when $\bar{T} < 2$); and vice versa for trial periods below \bar{T} . The threshold value \bar{T} rises when the values of $A, \alpha s, s_0$ and t_0 are larger, and δ_m is smaller. Highly novel software have $\bar{T} = 2$.
- (ii) Let incremental learning be substantial (λ large), and the sample is hard to reinstall (d small). The

monopolist is better off precluding reinstallation and extending the maximum trial period beyond \bar{T} (when $\bar{T} < 2$) rather than giving a free sample with possible reinstallations of unitary length; and vice versa for trial periods below \bar{T} . The threshold value \bar{T} drops when the values of $A, \alpha s, s_0$ and t_0 are larger, and δ and δ_m are smaller. Very novel software have $\bar{T} = 1$.

Proof See Appendix B: Case 2. □

This last proposition offers important insights on the need for the monopolist to minimize reinstallation of free software sample. Proposition 8 features cases where reinstalling a free sample may turn out to be a desirable property from the monopolist’s standpoint.²⁵ Case (i) states that when incremental learning is low and the software is conventional/basic/modular, the monopolist should maintain a unitary length for the maximum evaluation period and not preclude reinstallation. In contrast, if the monopolist were to extend the evaluation period past \bar{T} and fully preclude reinstallation, he would then postpone potential revenues that could have been received at the end of the unitary trial period. Furthermore, due to the low incremental learning, the additional revenues generated by an extension of the trial time would be insufficient to compensate for the postponement of the original revenues. Obviously, this outcome arises even more naturally when the monopolist is impatient (δ_m relatively small).

²⁵ The proof in Appendix B assumes that consumers cannot buy beyond period 2. Given the nature of scenario (1), the optimal solutions for trial time and proportion of features remain identical as in the basic model.

On the other hand, case (ii) states that, when the product is novel/complex/integrated/high-intensity and has a short life cycle, it is sensible for the monopolist to extend the initial evaluation period beyond \bar{T} and fully prevent reinstallation. This is true when incremental learning is high. Since reinstallation is assumed hard, extending the evaluation period beyond \bar{T} is a better means of rendering learning more effective. This generates greater revenues as compared with the case where reinstallation is allowed. As the monopolist become more impatient and the product more innovative and complex/integrated/high-intensity, it is not necessary to extend the trial period much beyond the unitary period.

In general, given that the observed majority of evaluation periods for shareware are from one to three months, we should expect the discount rate for both monopolist and consumers to be fairly close to one, based on that short time span. In that case, Proposition 8 points to scenario 2 of no reinstallation as the best option for the monopolist.

Notwithstanding the above discussion, it is clear that the overall best-case scenario for the monopolist is to allow reinstallation only so far as to capture the full effect of learning, and then stop the reinstallation process from that point on. Our reinstallation scenario 1 is justifiable when there is uncertainty about the dispersion of learning speed/abilities in the population, and more importantly, when the cost of enforcing the evaluation agreement is large, as may be the case for mass consumer software markets. Indeed integrating this fixed cost in our analysis would have made the outcome of scenario 1 of sample reinstallation even stronger.

6 Conclusions

One common approach used to market software products is to offer free samples or shareware. However, the design of shareware is a non-trivial task when the sample can potentially cannibalize the sales of the original product. We introduce a new software classification scheme according to shareware characteristics that help assess the complexity of use and novelty of the product among other properties. For each type of software, we identify the optimal combination of trial time and proportion of features that should be included in the free samples. Our theoretical results match the practices followed by a wide array of software vendors, who use free samples as a marketing tool. These results can also serve as a normative guide for the design of new shareware.

Furthermore, we show that when consumers' learning is progressive, it may be beneficial for the monopolist *not to deter* reinstallation to capture a new fringe of customers while still obtaining revenues from customers who were willing to buy the commercial product *early on*. For software products that are conventional, basic and modular, this strategy is better than extending the trial time and preclude reinstallation, which would delay revenues from all buyers. On the other hand, it makes sense for the monopolist to fully prevent reinstallation for novel and complex products and to extend the trial time duration to facilitate greater learning.

In this version of our model, trial time and proportion of features were considered substitutes in the learning process. Future research will examine the case of trial time and proportion of features being complements. Complementarity means that more features and more trial time given together tend to enhance sales. Studying the case of features that are more or less desirable for a segment of the population, and where learning may have an adverse impact on sales could also add a new twist to our model. Finally, analyzing the impact of simultaneously having an old and new generation of products, and/or relaxing the monopolistic framework are other logical avenues for extensions of our model.

Appendix A: Basic model proofs

Below, we analyze the Subgame Perfect equilibria of the basic game. Using Proposition 1 we can rule out as an equilibrium the case where a monopolist would offer a free sample and consumers decline using the free sample and buy the product right away. One interesting insight from this proposition is that the payoff to the consumer from buying without receiving a sample or buying while declining to use the free sample must be the same.

Case 1 (A free sample is offered) The solution (s^*, t^*) is such that $s_0 \leq s^* \leq 1$ and $t_0 \leq t^* \leq 1$.

By backward induction, the consumers who use the sample and then buy (B) must have their payoffs satisfy: (1) $\delta U - \delta P + \delta^2 \theta V / (1 - \delta) > \delta U + dUV\delta^2 / (1 - \delta)$ (Sampling and Buy > Sampling and Reinstall Sample). Condition (1) is equivalent to $\theta > (1 - \delta)P / \delta K(s, t)$; where $K(s, t) = [1 - ds ((1 + t)/2)]V > 0$. The demand function is given by $D(P, t, s) = N[1 - \Psi((1 - \delta)P/\delta K)]$.

The monopolist's profit is $\Pi_1 = D(P, t, s) \times [P - c] - F - N\gamma c$. The optimal solution for the price P_1^* satisfies the following first order condition:

$$\frac{\partial \Pi}{\partial P} = \frac{\partial D}{\partial P} [P - c] + D = 0 \tag{1}$$

Solving Eq. 1 leads to:

$$P_1^* = \frac{c}{2} + \frac{\delta K^* \theta_H}{2(1 - \delta)} \tag{2}$$

To insure that any demand exists, we check that $\theta_H > (1 - \delta)P_1^*/\delta K^*$, which is implied by $\theta_H > ((1 - \delta)/\delta)c$ when $K^* > 1$; where the variable K^* represents the function $K(\cdot)$ evaluated at the point (s^*, t^*) . The monopolist's optimal profit is $\Pi_1^* = (N/(\theta_H - \theta_L)) \times (\delta K^*/4(1 - \delta)) \times [\theta_H - (((1 - \delta)c)/\delta K^*)]^2 - F - N\gamma c$.

The optimal proportion of features s^* is given by the following necessary first order conditions depending on whether the solution is a corner or interior one:

$$\left\{ \begin{array}{l} s^* = s_0 \\ s^* = \bar{s} \\ s^* = 1 \end{array} \frac{\partial \Pi}{\partial s} = \frac{\partial D}{\partial s} [P - c] \right\} \begin{array}{l} < 0 \\ = 0 \\ > 0 \end{array} \tag{3a, 3b, 3c}$$

with $\frac{\partial D}{\partial s} = N \times \frac{P}{K^2} \times \Psi'(\cdot) \times \frac{\partial K}{\partial s}$. Thus, Eqs. 3a, b, c become (given t^*):

$$\left\{ \begin{array}{l} s^* = s_0 \\ s^* = \bar{s} \\ s^* = 1 \end{array} \frac{\partial K}{\partial s} = -d \left(\frac{1+t^*}{2} \right) V^* + \left[1 - ds \frac{1+t^*}{2} \right] \right. \tag{4a}$$

$$\times \frac{\partial V}{\partial s} \left\{ \begin{array}{l} < 0 \\ = 0 \\ > 0 \end{array} \right. \tag{4b, 4c}$$

Similarly, the optimal trial time t^* is given by the following first order conditions:

$$\left\{ \begin{array}{l} t^* = t_0 \\ t^* = \bar{t} \\ t^* = 1 \end{array} \frac{\partial \Pi}{\partial t} = \frac{\partial D}{\partial t} [P - c] \right\} \begin{array}{l} < 0 \\ = 0 \\ > 0 \end{array} \tag{5a, 5b, 5c}$$

Again, Eqs. 5a, b, c become (given s^*):

$$\left\{ \begin{array}{l} t^* = t_0 \\ t^* = \bar{t} \\ t^* = 1 \end{array} \frac{\partial K}{\partial t} = -\frac{ds^*}{2} V^* + \left[1 - ds^* \frac{1+t}{2} \right] \frac{\partial V}{\partial t} \right\} \begin{array}{l} < 0 \\ = 0 \\ > 0 \end{array} \tag{6a, 6b, 6c}$$

Since fixed costs are assumed *independent* of the sample's actual values for s and t , our (local) optimal proportion of features and optimal trial time are determined at the margin by maximizing static demand. Maximizing demand in our framework

translates into maximizing the function $K(\cdot)$, which pins down the trade-off between the impact of learning on the desirability of purchasing the product versus the attractiveness of sample reinstallation.

Thus, our solutions are maximizing the function $K(\cdot)$ given the parameter values. Regarding interior solutions, a series of simple algebraic manipulations respectively using (4b) and (6b) leads to:

$$\bar{s}(t^*) = \frac{1}{d(1+t^*)} - \frac{(1-\alpha)}{2\alpha} t^* - \frac{1}{2\alpha A} \tag{7}$$

$$\bar{t}(s^*) = \frac{1}{ds^*} - \frac{\alpha}{2(1-\alpha)} s^* - \frac{1+A(1-\alpha)}{2A(1-\alpha)} \tag{8}$$

It is important to note that the proportion of features s^* and the trial time t^* cannot jointly be *interior* solutions at the same time, since we can easily show that the Hessian is not negative semi-definite at the point (\bar{s}, \bar{t}) . In fact, the pair (\bar{s}, \bar{t}) represents a saddle point solution in the feasible space.

Case 1.1 $(s^*, t^*) = (1, 1)$ will be true if (4c) and (6c) hold.

It is easy to see that (4c) is equivalent to $1 < \bar{s}(1)$ and (6c) is equivalent to $1 < \bar{t}(1)$. These two conditions will be satisfied when d is close to zero (reinstallation cost is infinite).

Case 1.2 $(s^*, t^*) = (s_0, t_0)$ will be true if (4a) and (6a) hold.

Both conditions are equivalent to having $s_0 > \bar{s}(t_0)$ and $t_0 > \bar{t}(s_0)$. These conditions will be satisfied even when d is large (reinstallation cost is low) but not arbitrarily close to 1; α is in a compact range (not too close to either 0 or 1); and some or all of the following hold: A, s_0 and t_0 are large. Moreover, when A is small then α must be large, and vice versa.

Case 1.3 $(s^*, t^*) = (1, t_0)$ will be true if (4c) and (6a) hold.

These conditions are equivalent to $1 < \bar{s}(t_0)$ and $t_0 > \bar{t}(1)$ as well as $(1 - \alpha)(1 - t_0) < \alpha$. These conditions are satisfied if α is large, t_0 and d small; and both A and s_0 are either large or small in tandem (d must be close to 0 when A and s_0 are both small). The conditions A and s_0 being either large or small together insure that $K(1, t_0) \geq \max\{K(s^*, t^*); s^* \in \{s_0, \bar{s}, 1\}; t^* \in \{t_0, \bar{t}, 1\}\}$.

Case 1.4 $(s^*, t^*) = (s_0, 1)$ will be true if (4a) and (6c) hold.

These conditions are equivalent to $s_0 > \bar{s}(1)$ and $1 < \bar{t}(s_0)$ implying $2(1 - \alpha) > \alpha s_0$. These conditions are satisfied when α and s_0 are small. A higher value for A will make the condition on α less binding.

Case 1.5 $(s^*, t^*) = (\bar{s}, t_0)$ will be true if (4b) and (6a) hold. The latter condition is equivalent to $t_0 > \bar{t}(\bar{s})$ and both conditions imply that $(1 - \alpha)(1 + t_0) < \alpha\bar{s}$. This will be true when α and A are large and s_0 and d are small. The condition s_0 small is necessary otherwise, case 1.2 would dominate.

Case 1.6 $(s^*, t^*) = (s_0, \bar{t})$ will be true if (4a) and (6b) hold. The first condition is equivalent to $s_0 > \bar{s}(\bar{t})$ and both conditions imply $(1 - \alpha)(1 + \bar{t}) > \alpha s_0$. This is true when α is small, A and s_0 are large and d is large enough.

Case 1.7 (Not a solution) $(s^*, t^*) = (\bar{s}, 1)$ will be true if (4b) and (6c) hold. This pair *cannot be a solution*. The latter condition is equivalent to $\bar{t}(\bar{s}) > 1$ and both conditions imply that $(1 - \alpha)(1 + t_0) > \alpha\bar{s}$ which will be true if α is small, A and t_0 large, with d not too small. This solution is dominated by (s_0, t^*) with $t^* \in \{t_0, \bar{t}, 1\}$. The idea being that if A is large and α is small, reducing the sample features does not reduce its marketing power, while at the same time it prevents sample re-use.

Case 1.8 (Not a solution) $(s^*, t^*) = (1, \bar{t})$ will be true if (4c) and (6b) hold. Again, this pair *cannot be a solution*. The first condition is equivalent to $\bar{s}(\bar{t}) > 1$ and both conditions imply that $(1 - \alpha)(1 + \bar{t}) < \alpha$; which will be true if α and A are large and d small. This solution is dominated by picking (s^*, t_0) with $s^* \in \{s_0, \bar{s}, 1\}$. The idea being that if α is large, trial time plays a minor role, and hence reducing it does not lower the attractiveness of the product since A is large.

Case 2 (No Free Sample is Given) Assume that the monopolist strategy space is limited to picking a proportion of features $s < s_0$. Since the proportion of features is below the minimum, then $V = 1$. Thus, the function $K = 1 - ds((1 + t)/2)$ takes values in the interval $[0, 1]$. It is easy to see that the demand $D(P, t, s)$ is decreasing in s and t . By not distributing a free sample, the monopolist is also not incurring the cost $N\gamma c$ of making and distributing copies of the sample. Thus, the optimal response for the monopolist is to set s^* and $t^* = 0$. Moreover, the standard first order condition for selecting the optimal price P_2^* leads to $P_2^* = (c/2) + (\delta\theta_H/2(1 - \delta))$, and the profit is $\Pi_2^* = N/(\theta_H - \theta_L)(\delta/4(1 - \delta))[\theta_H - ((1 - \delta)/\delta)c]^2 - F$. Lastly, for the demand to exist we need $\theta_H > ((1 - \delta)c/\delta)$.

We need to check that selecting s^* and $t^* = 0$ is optimal over the *global* strategy space. This will be the case if $\Pi_2^* > \delta_m \Pi_1^*$ where Π_1^* is the maximum profit generated in Case 1. It is easy to check that $\Pi_2^* > \delta_m \Pi_1^*$ when γ large, δ_m small and $K^* < 1$. The latter will be

true when the learning parameter A is small and/or s_0 is close to 1, and/or t_0 is close to 1, and/or the reinstatement cost is low (d large), and/or α is small. Otherwise, the solutions $s_0 \leq s^* \leq 1$ and $t_0 \leq t^* \leq 1$ explored in Case (1) are optimal over the *global* strategy space, i.e. including selecting $s < s_0$ and $t < t_0$, under contrary assumptions. \square

Appendix B: Multi-period learning proofs

Case 1 (2-period learning) We assume that in addition to the strategies that were available before, consumers can also buy the product in period 2; but not beyond that period. Below, we analyze the equilibrium where two sequential market demands co-exist with one segment of consumers who prefer buying at the end of period 1, and another segment who will buy after two periods.

Subcase 1.1 The solution (s^*, t^*) is such that $s_0 \leq s^* \leq 1$ and $t_0 \leq t^* \leq 1$.

To keep the notations lighter we examine the consumers payoffs from the standpoint of present values at the *end* of period 1 (after sampling), and not the beginning of stage 1 as was done in the basic model. By backward induction, the consumers who will use the sample and then buy (B) at end of period 1 must have their payoffs satisfy the following conditions; given that learning occurs according to the learning function L_1 :

- (1) $U - P + \delta\theta V + \delta^2\theta\lambda V/(1 - \delta) > U + \delta dUV + \delta^2 dU\lambda V/(1 - \delta)$ (Buy at end period 1 > Not-Buy)
- (2) $U - P + \delta\theta V + \delta^2\theta\lambda V/(1 - \delta) > U - \delta P + \delta dUV + \delta^2\theta\lambda V/(1 - \delta)$ (Buy at end period 1 > Buy at end of period 2)

By backward induction, consumers who will use the sample and then buy (B) at end of period 2 must have their payoffs satisfy:

- (3) $U - \delta P + \delta dUV + \delta^2\theta\lambda V/(1 - \delta) > U + \delta dUV + \delta^2 dU\lambda V/(1 - \delta)$ (Buy at end period 2 > Not-Buy)

Note that conditions (2) and (3) imply condition (1).

Condition (3) is equivalent to $\theta > \bar{\theta}_1 = ((1 - \delta)P)/\lambda\delta K(s, t)$; where $K(\cdot)$ is defined as before in the basic model. Condition (2) is equivalent to $\theta > \bar{\theta}_2 = (1 - \delta)P/\delta K(s, t)$. It is easy to show that $\bar{\theta}_2 > \bar{\theta}_1$ since $\lambda > 1$. More importantly, the segment of consumers that satisfy $\theta > \bar{\theta}_2$ will buy at the end of period 1, and the segment of consumers satisfying $\bar{\theta}_2 > \theta > \bar{\theta}_1$, will buy in the second period. The demand from consumers buying in period 1 is $D_1(P, t, s) = N[1 - \Psi(\bar{\theta}_2)]$. The

residual demand from consumers buying in period 2 is $D_2(P, t, s) = N[\Psi(\bar{\theta}_2) - \Psi(\bar{\theta}_1)]$. The monopolist’s profit (valued at end of period 1) is $\Pi = [D_1 + \delta_m D_2] \times [P - c] - F - N\gamma c$; which is the present value of profits obtained from the total demand. Let us denote the total demand by $D = [D_1 + \delta_m D_2]$. The optimal solution for the price P^* is found using the following first order condition:

$$\frac{\partial \Pi}{\partial P} = \frac{\partial D}{\partial P} [P - c] + D = 0 \tag{9}$$

Solving Eq. 9 leads to:

$$P^* = \frac{c}{2} + \frac{\delta K^* \theta_H}{2(1 - \delta)} \times \frac{\lambda}{\lambda - \delta_m(\lambda - 1)} \tag{10}$$

Given the expression for the total demand and the price above, the monopolist’s optimal profit is given by $\Pi^* = (N/(\theta_H - \theta_L)) \times (\delta K^*/(4(1 - \delta))) \times (\lambda/(\lambda - \delta_m(\lambda - 1))) \times [\theta_H - (\lambda - \delta_m(\lambda - 1))]/\lambda \times ((1 - \delta)c)/\delta K^*]^2 - F - N\gamma c$.

Just as was done in Appendix A, it is straightforward to show that maximizing the profit function is equivalent to maximizing the function $K(\cdot)$ with respect to trial time and proportion of features. Therefore, the solutions (s^*, t^*) are the same as the ones described in Appendix A.

Subcase 1.2 The solution (s^*, t^*) is such that $s^* < s_0$ and $t^* < t_0$. This sub-case is the same as in Appendix A.

Case 2 (One-period learning; no reinstallation possible) Since the sample is essentially non-durable it is straightforward to show that the optimal solution $(s^*, t^*) = (1, T)$. The consumers that sample and buy must satisfy $U - P + \delta^T \lambda^{T-1} V_{\max} \theta / (1 - \delta^T) > U$; since no reinstallation is possible; and given that learning occurs according to the learning function L_2 : where $V_{\max} = A(\alpha + (1 - \alpha)T) + 1$. Maximizing profit with respect to price gives:

$$P_3^* = \frac{c}{2} + \frac{\lambda^{T-1} \delta^T V_{\max} \theta_H}{2(1 - \delta^T)}, \tag{11}$$

Again, given the assumption that completely preventing the unauthorized reinstallation of free samples is costless for the monopolist (independent of d), the monopolist’s optimal profit must given by $\Pi_3^* = (N/(\theta_H - \theta_L)) \times (\lambda^{T-1} \delta^T V_{\max} / (4(1 - \delta^T))) \times [\theta_H - ((1 - \delta^T)c)/\lambda^{T-1} \delta^T V_{\max}]^2 - F - N\gamma c$. Comparing Case 1 with Case 2, we surmise that Case 1 will be favored by the monopolist when $\delta_m \Pi^* > \delta_m^T \Pi_3^*$. This is true when the following sufficient condition holds:

$$\frac{\delta K^*}{(1 - \delta)} \times \frac{\lambda}{\lambda - \delta_m(\lambda - 1)} > \frac{\lambda^{T-1} \delta^T V_{\max}}{(1 - \delta^T)} \tag{12}$$

Let $\bar{T}(A, d, \alpha, t_0, \delta, \delta_m, \lambda)$ be such that:

$$\frac{\lambda^{\bar{T}-1} \delta^{\bar{T}} \bar{V}_{\max}}{(1 - \delta^{\bar{T}})} = \frac{\delta K^*}{(1 - \delta)} \times \frac{\lambda}{\lambda - \delta_m(\lambda - 1)} \tag{13}$$

A solution \bar{T} exists if the RHS of (12) is monotonic in T , and under other conditions on the parameters defined below. Thus (12) is equivalent to:

$$\frac{\lambda^{\bar{T}-1} \delta^{\bar{T}} \bar{V}_{\max}}{(1 - \delta^{\bar{T}})} > \frac{\lambda^{T-1} \delta^T V_{\max}}{(1 - \delta^T)} \tag{14}$$

From Eq. 14, we can see that Case 1 will be favored when $T \geq \bar{T}(A, d, \alpha, t_0, \delta, \delta_m, \lambda)$ as long as $\partial(\lambda^{T-1} \delta^T V_{\max} / (1 - \delta^T)) / \partial T < 0$. It is easy to check that this latter derivative being negative is implied by λ close to 1. Moreover, these conditions with δ_m small and d large are also sufficient for the value of $(\delta K^* / (1 - \delta)) \times (\lambda / (\lambda - \delta_m(\lambda - 1)))$ to be contained in the interval of minimum and maximum values for the function $(\lambda^{T-1} \delta^T V_{\max} / (1 - \delta^T))$. Furthermore, basic comparative statics on Eq. 13 reveals that \bar{T} must be increasing in A, α, s_0, t_0 ; since d is assumed large. However, the effect of raising the discount rate δ on \bar{T} is ambiguous.

On the other hand, $\partial(\lambda^{T-1} \delta^T V_{\max} / (1 - \delta^T)) / \partial T > 0$, implies that Case 1 will be favored when $\bar{T}(A, d, \alpha, t_0, \delta, \delta_m, \lambda) \geq T$. It is easy to check that this latter derivative being positive is implied by λ large. Moreover, these conditions with δ_m small and d small are also sufficient for the value of $(\delta K^* / (1 - \delta)) \times (\lambda / (\lambda - \delta_m(\lambda - 1)))$ to be contained in the interval of minimum and maximum values for the function $\lambda^{T-1} \delta^T V_{\max} / (1 - \delta^T)$. The comparative statics from (13) follows from the same argument as before. It shows that \bar{T} must be decreasing in A, α, s_0 and t_0 , and increasing in δ ; especially when d is assumed small. □

References

1. G.E. Belch and M.A. Belch, *Introduction to Advertising and Promotion Management*, Richard D. Irwin, Homewood, IL (1990).
2. M. Cakanyildirim and T. Dalgic, Using demonstration to promote information products, working paper, University of Texas at Dallas (2002).
3. S.-Y. Choi, D.O. Stahl and A.B. Whinston, *The Economics of Electronic Commerce*, McMillan Technical Publishing (1997). Available at: <http://www.smartecon.com/products/catalog/eecflyer.asp>

4. R.H. Coase, Durability and monopoly, *Journal of Law and Economics* 15 (1972) 143–149.
5. A. Dhebar, Durable-goods monopolists, rational consumers, and improving products, *Marketing Science* 13(13) (1994) 100–120.
6. E. Haruvy and A. Prasad, Optimal freeware quality in the presence of network externalities: an evolutionary game theoretical approach, *Journal of Evolutionary Economics* 11 (2001) 231–248.
7. A. Heiman and E. Muller, Using demonstration time to increase new product acceptance: controlling demonstration time, *Journal of Marketing Research* 33 (1996) 422–430.
8. A. Heiman and E. Muller, The effects of competition and costs on demonstration strategies in the Software Industry, *Mimeo*, Tel Aviv University (2001).
9. M.L. Katz and C. Shapiro, Network externalities, competition, and compatibility, *American Economic Review* 75(3) (1985) 424–440.
10. G.C. Parker and M.W. Van Alstyne, Information complements, substitutes, and strategic product design, working paper, Economics Dept., University of Michigan (2000).
11. N. Stokey, Rational expectations and durable goods pricing, *Bell Journal of Economics* 12 (1981) 12–128.
12. S. John, Vertical product differentiation, *AER Papers and Proceedings* (1986) pp. 393–398.
13. L.N. Takeyama, Strategic vertical differentiation and durable goods monopoly, *Journal of Industrial Economics* 50(1) (2002) 43–56.
14. J. Tirole, *The Theory of Industrial Organization*, MIT Press, Cambridge Mass. (1988).