

## MODELING DYNAMIC SOFTWARE SAMPLING STRATEGIES

Yanbin Tu, University of Connecticut, USA  
Min Lu, University of British Columbia, Canada

### ABSTRACT

Previous studies show that product sampling is one of the effective marketing strategies to stimulate the real time and future consumption. In order to increase software sales, many software vendors use the online product sampling strategy, i.e., releasing free online samples of their software products to consumers for trial. Consumers can freely download and try the lower version of software without a time limitation or they can freely access and try the full version of the software as Demo edition within a limited time period. Product sampling for information goods like software has some advantages over that for some physical products like pizza. Firstly, unlike pizza sample, which Pizza Hut needs flour, cheese and meat to make, the software vendor incurs nearly zero marginal cost to make a software sample. Secondly, for physical products, the product sampling strategy is usually only applied to non-durable products such as shampoo. As information good, software is durable good. The software vendor needs to supply only one copy of the software to the sampler for whatever length period trial. Since there is a substitution or cannibalization effect between the product and the product sample, non-durable physical product sample attains the whole quality of the product, but has restriction on quantity. For example, shampoo producer gives only 5 bags of shampoo with full quality as product sample to one household. Generally, in order to avoid the substitution or cannibalization effect between the sample software and the original software itself, software sampling is restricted either in its quality (functionality) or in its time length of legitimate usage. Therefore, we have two types of sampling strategies for software: the low-quality sampling strategy and the limited-time-trial sampling strategy.

Generally, we can divide the online information good sampling into two categories: Static sampling such as music sampling and dynamic sampling such as software sampling. In static sampling, a consumer tries information goods via the Internet. For example, music samples at Amazon.com or audile book at Audible.com. The sampler derives some utility from sampling, then assign himself an expected utility for the real product. He will compare his expected utility with the information good price and makes a purchase decision. It is a static sampling because the consumer will not try the sample again unless he takes the sample as the substitute of the original product. Static sampling is usually applied to experience information good such as digital music.

The process of dynamic information good sampling is different. A consumer tries information good sample, but he neither buys the information good nor gives up sampling right away. He will try the sample again afterwards. In this way, as he learns more and more about the information good, he might accumulate his goodwill to buy. Once his goodwill is high enough, he will choose to buy the information good, or, if he losses his interest, he will drop the sampling process. In the case of limited-time-trial sampling, if the sampler's goodwill has not been accumulated high enough to buy the product at the expiry date, he has no more chance to try it again, and the sampling process is over.

Software sampling is a dynamic sampling process because it's impossible to know all of the functionality of the software after only one trial. This study models two dynamic software sampling strategies. We try to answer the following questions which have not been thoroughly explored so far: For the software vendor, how to decide the optimal sample quality under low-quality sampling strategy or how to choose the legitimate trial period for limited-time-trial sampling strategy so that he can collect the maximum profits in the long run? Which factors and how they affect the software vendor's revenue through the product sampling strategy? This paper shows that, for low-quality sampling strategy, the optimal quality level of the software is determined by the likelihood of the sampler being a free-rider. In general, the faster the sampler learns the software from sampling, the higher total revenue the software vendor can collect. And, the quicker the sampler forgets, the less total revenue the vendor can obtain. For the limited-time-trial sampling, our numerical analysis demonstrates that the optimal trial period is negatively related to the peak time of the sample life cycle. Given the learning speed, the optimal trial period is negatively related to the forgetting speed.

References available upon request