Persistence Effects in a Dynamic Discrete Choice Model. 
Application to Low-End Computer Servers.

Szabolcs Lőrincz
University of Toulouse
szabolcs.lorincz@univ-tlse1.fr

To be competing for the Young Economist Award.

March 14, 2005
Abstract

I introduce preference persistence into a dynamic discrete choice model of demand for durables. This persistence may arise, for example, when the products can be categorized into a few number of formats, which involve special knowledge, maintenance and upgrade. The standard optimal stopping problem of when to buy a new product (Rust (1987), Melnikov (2000)) is completed by the upgrade problem: customers who already have a product may choose to upgrade it, but this upgrade is format specific. Hence, the expected future upgrade qualities of different formats must be taken into account already at the purchase decision of a new product. I estimate my model on a data set of low-end computer servers, where formats are represented by operating systems. Results suggest that the model is better able to capture main tendencies in the segment than a static or a simple optimal stopping model.

Keywords: preference persistence, differentiated durables, Markov decision process, computer servers

JEL classification: C33, D12, D91, L63
1 Introduction

The choice of a specific durable good very often includes the choice of a more broadly defined product platform or format. For example, in many cases the purchase of a computer is also a decision about its operating system. Even though brands and products might be pretty differentiated from each other in many other respects, a commonly shared platform brings them much closer in the view of a consumer. Maintenance and product upgrades can be platform-specific. Also, ‘applications’ might be tied to formats: for instance, software codes written for a given operating system might not run on a different OS. In addition, to use a product efficiently the consumer must learn it, and this knowledge can be vastly different across platforms. As a result, by purchasing a good the consumer might end up being locked into its format: Even if later she decides to replace her existing durable and also to change platform, this will already involve a switching cost of learning the new platform. So, the consumer has a motivation of keeping the platform previously chosen. In other words, technology might induce some persistence in preferences.

I propose introduction of format specific preference persistence in a dynamic model of demand for differentiated durable goods. I start from recent results in the econometric modeling of Markovian discrete decision processes (DDPs). Rust (1987) constructs a dynamic logit model describing replacement decision of a durable good. This is an optimal stopping problem where the agent must decide on the optimal time of purchase. Melnikov (2000) expands this frame to model choice from a set of differentiated durable goods whose quality improves stochastically over time. In my model, I couple a Melnikov-type optimal stopping problem with a persistence effect. The consumer makes her choices between several differentiated durables which can be partitioned into a few number of formats. A consumer who already has a good can upgrade it, which means an improvement or development, without scrapping the old product. However, this upgrade is format specific and this is the root of preference persistence.

So, dynamics is generated by two sources in my model. First, durability of the good creates a standard optimal stopping problem. The customer faces a trade-off when deciding about the optimal time to buy a product, when quality of new products improves stochastically over time: She can purchase and get utility in the current period or she can postpone this decision for a later time when better quality products may be available. Second, a customer who already has a product can choose between simply using her original product and between a format specific upgrade. Hence, expected future upgrade qualities of different formats must be taken into account already at the stage when the purchase decision about a new product and, hence, about its format is being made.

I apply the model to the case of low-end server computers where formats are represented by
operating systems. Servers are computers connecting client computers, which are most often PCs, of a network and provide them file and print sharing, authentication, data storage and access to specialized software. In many cases, servers are operated on a 24 hours base, hence reliability and security is essential. To maintain these features, software and hardware upgrades are often carried out. The most important piece of any computer’s software code is the operating system, which translates user commands for the computer and interprets messages from the machine. The choice of OS is crucial not only from the reliability and security point of view but also because OSs can be very different and may require specialized knowledge of the user. In addition, the quality and nature of both, new products and upgrades can be different for different OSs. These features motivate the use of my model for durable goods with preference persistence.

The class of DDPs, which this model belongs to, is surveyed by Rust (1994). In particular, I specify a dynamic nested logit model where nests represent formats, that is, different operating systems. Using a world-wide panel of server purchases, I propose a dynamic extension of the sequential estimation procedure of McFadden (1981): First, static conditional logit models of nests are estimated. Second, I estimate transition probabilities of inclusive values from the first step. Finally, a dynamic logit model of choice between nests is set up, where utility of a nest is a function of its inclusive value. In this step, I obtain maximum likelihood estimators using the nested pseudo likelihood (NPL) algorithm of Aguirregabiria and Mira (2002).

I compare the model to two benchmark models: One is a static nested logit model and the other is a simple optimal stopping model without persistence effects. The results show that the full model is better able to explain dynamics of demand for servers: The rising share of systems with Linux and Windows, and the decline in NetWare OS figures. This validates the forward looking assumption on consumer behavior and further suggests that persistence effects are indeed present and play an important role in customer choice.

The paper is built up as follows. Section 2 reviews related literature. Section 3 sets up the dynamic discrete choice model. Section 4 discusses industry, data and econometric specification. Section 5 presents empirical results and Section 6 concludes.

2 Related literature

To put my work in context, I provide a (necessarily short and incomplete) review of the empirical literature on dynamic discrete choice models. First, I list some important applications categorized by their sources of dynamics modeled. Then I briefly turn to the literature on related solution and estimation problems.
2.1 Motivation: from static to dynamic models of discrete choice.

The literature on static models of demand for differentiated products has seen a huge development in the recent decades. McFadden (1981), Berry (1994), Berry, Levinsohn and Pakes (1995) have established a modeling framework including the popular logit, nested logit, and random coefficients models. Products are treated as portfolios of characteristics with heterogeneous consumers choosing the alternative providing the highest utility. Consumer heterogeneity is mainly modeled by assuming functional forms describing unobserved components of preferences. Static framework means that consumers do not consider effects of past and future states of world when choosing their preferred alternative in the present. In practice, this means that estimating models from panel data assumes consumers making their decisions each period recurrently, regardless of their choices in other periods.

Of course, the assumption of static optimizing behavior can be questioned in numerous cases. Accordingly, a significant portion of the literature has focused on expanding the static framework into a dynamic one. The source of dynamics can vary from application to application, hence the forms of models are more heterogenous. For example, when choice products are durables it is reasonable to assume that a consumer does not make a purchase each period since a durable good provides a flow of services to its buyer for more than one period. This leads to an optimal stopping problem, i.e., the choice of optimal purchase time. In a seminal paper, Rust (1987) sets up a dynamic version of McFadden’s logit model to describe and analyze empirically the problem of an agent who chooses optimal time of replacing a used durable, which is otherwise non-differentiated. Melnikov (2000) further expands this model to the case of explicit product differentiation. His model enables him to analyze the effect of technological progress, which affects product quality. This is very important for most modern durable goods such as computers, printers, cars, etc., whose quality is rapidly improving over time.

Dynamics can be important for non-durables too. For example, Hendel and Nevo (2003) model demand for storable differentiated goods, washing detergents. Since the product can be stored, consumers adjust their shopping time to producers’ price reductions (sales). Hence, the way consumers form expectations about these sales is crucial and brings rich dynamics into demand patterns, even if consumption is smoothed. Ackerberg (2003) studies consumer choice of experience goods. Here, the process which links decisions in different time periods is consumers’ learning about a new product. This process is largely affected by producers’ advertizing behavior. These effects can be analyzed by the structural model.

Another important source of dynamics can be network externality. In many cases, for a consumer who is choosing between platforms, or formats, of products the number of other consumers choosing a given platform, directly or indirectly, may yield a positive utility value for that
platform. Hence, dynamics is generated by the fact that a utility maximizing consumer must take into account expected future number of purchasers of each formats, already at the period of choice between these platforms. For example, Park (2004) sets up a model of consumer discrete choices and producers pricing for video players. The platforms are VHS and Beta, indirect network externalities play role through the number of available prerecorded video movies: The more consumers choose a platform the larger variety of movies will be profitable for movie sellers on this format. Increased variety, in turn, represents a utility value for the consumer. Another example might be the demand for personal computer operating systems: The more people choose a given format (OS) the more application software code will be written on it, which, again, is valued by consumers.

The model I present in this paper is closely related to the work of Rust and Melnikov. I add a persistence effect to their optimal stopping model. This persistence in preferences can be the result of learning. However, unlike Ackerberg, and mainly due to data limitations, I do not model explicitly the process of learning. Another interesting issue is the possible existence of indirect network externalities in the case of server OSs. I argue that, unlike for PC OSs, for server operating systems network externalities are weaker. For any server OS it is crucial that it could interoperate seamlessly with many applications. The incompatibility issue is relatively less important.\footnote{The incompatibility between PC and server OSs is an issue (see Subsection 4.1). However, the existing and strong network externalities on the market for PC OSs do not necessarily result in network externalities on the market for server OSs, or servers.}

\section{2.2 Computational and estimation problems}

The price of increased realism of empirical structural dynamic models of discrete choice, relative to the static framework, is increased computational complexity. Hence, the literature exploring estimation techniques for dynamic models has become very important. I mention two issues: Efficient solution and estimation of the underlying dynamic programming problems, and the reduction of choice spaces in discrete choice models.

Rust (1987) proposes a nested fixed point (NFXP) algorithm to estimate a class of discrete decision processes: In an inner iteration circle, for a given value of the parameter vector, it solves the dynamic programming model. In an outer circle, it searches the parameter vector which maximizes the log likelihood. Dynamic programming models may involve large computational burden due to the ’curse of dimensionality’, that is, the exponential increase in the size of state space as the number of state variables increases. Several alternative estimation methods have been proposed to reduce this burden. For example, Rust (1997) proposes randomization
to break the curse of dimensionality. Hotz and Miller (1993) introduce a simple nonparametric conditional choice probabilities (CCP) estimator. Aguirregabiria and Mira (2002) define a nested pseudo likelihood (NPL) algorithm, which 'swaps' the NFXP algorithm: In an inner circle, for a given solution of the dynamic programming model, it maximizes a pseudo likelihood function. In the outer circle, it updates solution of the dynamic problem. This outer circle corresponds to policy iteration, that is, Newton stepping, which is an efficient way of solving the Bellman fixed point problem. For the first outer iteration, NPL is the CCP estimator. When estimates are converged with several outer iterations, the estimator is equivalent to the NFXP MLE. For some models, however, computational burden can be much smaller. I use NPL to get maximum likelihood estimators of dynamic parameters of my model.

A complementary method to reduce computational complexity is to reduce choice spaces by inclusive values. Inclusive value of a discrete choice problem is the expected maximum utility from the choice set. For logit models, it has a simple analytic form. Using inclusive values as a method of replacing a more complex problem by several less complicated ones dates back to McFadden (1981). He proposes sequential estimation of static nested logit models by first estimating simple static conditional logit models of within nest choices, then to estimate a simple static logit model of choice between nests. In this latter, choice specific utilities are represented by functions of the corresponding inclusive values from the first step. In a dynamic framework, Melnikov (2000) uses the inclusive value of a simple conditional logit model to reduce the state space in an optimal stopping problem. Hendel and Nevo (2003) use inclusive values of product groups as sufficient statistics for price processes of these groups. They set up a dynamic model of consumer choice of storable goods when producers tend to reduce their prices significantly from time to time to boost sales. Modeling a similar situation as Hendel and Nevo, Aguirregabiria (2002) uses inclusive values in a dynamic discrete choice model as a method of valuing groups of products in a multi-stage budgeting framework. This branch of literature is closely related to my paper, which proposes basically a dynamic extension of McFadden’s sequential estimation principle of a nested logit model.

3 The model

I set up a dynamic model of discrete choice between durable goods, which are server computers in my application. At each period, a customer decides whether to make a purchase or not. If she does not have a product yet then she has an option to buy a new product or to postpone decision until next period. Since the good is durable it provides a flow of services not only in the period of purchase but all through its life-cycle. However, an old product can break down in each period with a given probability. Then, the customer finds herself again in the initial situation
of not having a product and deciding whether to make or not a new purchase. If a customer already has a product at the beginning of a period then, if it does not break down, she can keep it simply or she can upgrade it. Upgrade means an improving product update or expansion.

Dynamics is driven by two sources in this model. First, durability of the good creates an optimal stopping problem. At each period, a customer not owning any product faces a trade off: On the one hand, if she stops, that is, makes a purchase she starts enjoying the utility flow provided by the new product from the current period. On the other hand, if she postpones purchase until tomorrow then although she does not get the utility from the product today but next period she may purchase a technologically superior product, which yields her higher utility. Rapidly improving quality is a very common feature of durable information technology goods such as computers. I represent this phenomenon by an exogenous stochastic process for the level of product quality. The choice of upgrading creates a similar optimal stopping problem.

The second source of dynamics is a sort of preference persistence. Each product belongs to one, and only one, group (or nest). The nesting principle in this application is grouping by operating systems of server computers. When a customer upgrades her computer she can only choose an upgrade compatible with the operating system of her existing product. Hence, once the customer has a server she is already on a more determined path: The quality of her potential future upgrades depends on the quality process of upgrades available for the chosen nest. The customer is locked in the nest. Hence, already at the purchase decision of a new product expected future quality levels of its nest’s upgrades must be taken into account.

In the next two subsections I specify this decision problem as a dynamic nested logit model. First, I set up a general model where the customer forms expectations about each product’s future quality level separately. This model, however, has a too large dimension to be computationally feasible. So, in a second step I model a situation where the customer predicts only aggregate quality processes of nests, instead of those of individual products themselves. In this simplified model dynamic choice is between nests. Upon stopping, there is a standard static discrete choice problem of choosing an individual product from the chosen nest. This simplified model can already be the subject of structural econometric estimation.

### 3.1 A general dynamic nested logit model

A customer maximizes her lifetime utility by making a choice each period from a finite set of alternatives. Lifetime utility is a discounted sum of an infinite flow of instantaneous utilities:

\[
V(s_t) = \max_{\{j_\tau \in J_\tau\}_{\tau = t}^\infty} \left\{ E_t \left[ \sum_{\tau = t}^\infty \beta^{\tau-t} u_{j_\tau}(s_\tau) \right] \right\}.
\]
where \( u_j \) is the instantaneous, or static, indirect utility of choosing alternative \( j \); \( s_t \) is the value of a vector of state variables at period \( t \); and \( \beta \in (0, 1) \) is the discount factor. The set of alternatives available at a given period depends on the state: \( J_t = J(s_t) \). Evolution of the state vector is governed by a Markov-transition probability, which is assumed to be known by the customer: \( p(s_{t+1} | s_t, j) \).

Under certain regularity conditions, the problem can be represented by the Bellman equation of a stationary dynamic programming problem:

\[
V(s) = \max_{j \in J(s)} \left[ u_j(s) + \beta \int V(s') p(ds' | s, j) \right].
\]

Here, \( s' \) denotes the next period state vector. The intuition is that the problem of finding an infinite sequence of optimal choices is translated into a sequence of isomorphic one period optimization problems of finding the optimal choice in a given period assuming that future choices will be made optimally. That is why we can omit time subscripts and deal only with current and next period states, \( s \) and \( s' \), respectively.

The choice set \( J(s) \) can be partitioned into \( G+1 \) mutually exclusive subsets: \( J(s) = \bigcup_{g=0}^{G} g(s) \), with \( g(s) \cap g'(s) = \emptyset, \forall g, g' \) such that \( g \neq g' \). The first subgroup, denoted by \( g = 0 \), has one element and this is inaction, i.e., not buying any product. The other \( G \) subsets correspond to different operating systems, which is the nesting principle in this nested logit framework. One product can belong only to one subset hence knowing a choice \( j \) one can identify unambiguously the chosen subset \( g \) too.

The state vector has three components: \( s = (x, y, \varepsilon) \) and it is fully observed by the customer. First, \( x \) represents a set of product specific state variables, such as technical characteristics and price, each with a finite support. It has \( K+1 \) components from which \( K \) components are observed by the researcher and one is unobserved. Second, \( y \) is a customer specific state variable observed by the econometrician. Its support contains \( G+1 \) values: \( y \in \{0, 1, ..., G\} \). \( y = 0 \) means that the customer does not own any product at the beginning of current period, and \( y = g \) means that she already has a product at the beginning of current period and this product belongs to nest \( g \). Finally, \( \varepsilon \) represents customer and product specific heterogeneity, which is unobserved by the researcher.

Now we are ready to set up instantaneous indirect utility functions. There are four distinct cases according to observed customer specific states and choices:

Case 1. \( y = 0, g = 0, (j = g) \).

\[ u_j = c + \varepsilon_0. \]

Case 2. \( y = 0, g \in \{1, ..., G\}, j \in g \).

\[ u_j = x_j \gamma_g + \varepsilon_g + (1 - \sigma_g)\varepsilon_j. \]
Case 3. \( y \in \{1, ..., G\}, g = 0, (j = g). \)

\[ u_j = c_y + \epsilon^u_0. \]

Case 4. \( y \in \{1, ..., G\}, g \in \{1, ..., G\}, j \in g = y. \)

\[ u_j = x_j \gamma^u_g + \epsilon^u_g + (1 - \sigma^u_g)\epsilon^u_j. \]

Figure 1 displays graphically this preference structure: In Case 1, the customer does not own any product at the beginning of current period and she does not buy anything either. Her payoff is a constant \( c \), and her specific valuation \( \epsilon_0 \). In Case 2, she does not own any product at the beginning either, but now she opts for choosing a product from nest \( g \). Her payoff is the sum of a product specific value \( x_j \gamma^u_g \), where \( \gamma^u_g \) is a vector of parameters, and a composite unobserved-to-the-econometrician term. Case 3 is similar to Case 1: the customer does not buy anything. But, unlike in Case 1, now she already has a product, so she gets a format specific 'continuation value' \( c_y \). Finally, Case 4 shows payoffs in a situation where the customer already has a product and decides to 'upgrade' this product. This is represented by choosing an alternative \( j \) from the upgrade nest \( y \) of her original product. That is, she does not replaces her old product just makes an improvement on it. Note that for a customer who has a product, which belongs to a given nest it is not possible to choose an upgrade from an other nest. In my empirical application, nests corresponds to different operating systems. So, I assume that upgrades are operation system specific, which is certainly a good approximation in most real world cases.

The well known nested logit assumptions are used to describe distributions of unobserved heterogeneity terms \( \epsilon \), see, e.g., Cardell (1997): \( \epsilon_0 \) and \( \epsilon^u_0 \) are distributed identically and independently across all alternatives and periods with extreme value distributions. The terms \( \epsilon_g + (1 - \sigma_g)\epsilon_j \) and \( \epsilon_g \) are distributed identically and independently across nests and periods with extreme value distributions. The same is true for \( \epsilon^u_g + (1 - \sigma^u_g)\epsilon^u_j \) and \( \epsilon^u_g \). \( \sigma_g \in (0,1) \) governs within group correlation in nest \( g \). If its value converges towards 1 then for \( j \in g \) the correlation between the \( \epsilon_j \) terms converges towards 1. If \( \sigma_g \) goes to zero then the same correlation goes to zero too.

As a last step, we specify transition probabilities. I assume a form of conditional independence:

\[ p(x', y', \epsilon' | x, y, \epsilon, j) = h(\epsilon' | x', y')f(x' | x)l(y' | y, j). \]

First, next period values of unobserved heterogeneity terms \( \epsilon \) do not depend on current period states. This is the standard conditional independence assumption in dynamic discrete choice econometrics, see, e.g., Rust (1994). Second, product specific characteristics in \( x \) evolve as an
exogenous Markov-process. In particular, customer choices do not affect this quality generating process. Third, next period customer specific state \( y' \) depends on its own current value \( y \) and the current choice \( j \) as well. This is represented by the function \( l(y' | y, j) \):

\[
y' = \begin{cases} 
0, & \text{if } y = 0 \text{ and } j \in g = 0, \\
g, & \text{with probability } 1 - q, \text{ if } j \in g \neq 0, \text{ and/or } y = g \neq 0, \\
0, & \text{with probability } q, \text{ if } j \in g \neq 0, \text{ and/or } y = g \neq 0.
\end{cases}
\]

A customer who in the current period do not have any product \( (y = 0) \) and do not buy a new one either, will find herself in this same state at the beginning of next period \( (y' = 0) \). If she already has a product \( (y \neq 0) \) or she buys a new one in the current period then she will have this product at the beginning of the next period too, but only with probability \( (1 - q) \), where \( q \in (0, 1) \) is the exogenous probability of 'product break-down'. This probability represents a case where the existing product breaks down and, hence, it will not provide its service flow to the customer anymore who, as a result, is considered as not having a product (so, her state is reset at the beginning of this period to \( y' = 0 \)) and faces again the decision problem of buying or not a new product. Note that in the model, in each period, only those customers can buy a new product who have no product at the beginning of this period \( (y = 0) \). Those who already have one \( (y \neq 0) \) can only choose between not acting or upgrading from the same nest \( g = y \).

The set up of the general dynamic nested logit model is completed. For a given set of parameter values, its solution is a value function \( V(s) \) and a policy function \( j(s) \). The value function gives the expected discounted value of the optimal decision path if the process starts from state \( s \). The policy function gives the optimal decision in the current period if state is \( s \). Numerical solution of the problem is possible in principle. However, this is unfeasible in practice if the number of state variables is large (this is the curse of dimensionality) or if the number of alternatives is large. The next subsection specifies a simpler, but related problem whose size is smaller in both dimensions and, hence, its solution is computationally feasible.

### 3.2 A simplified dynamic nested logit model

I reduce state and choice spaces by transforming the general dynamic nested logit model into a simple dynamic logit model and \( G \) static conditional logit models of nests. In the simpler dynamic model, the choice is between nests. Here, instantaneous utilities of nests are represented by functions of inclusive values from static conditional logit models describing within nest choice of products. This works as follows.

Consider the case of a customer who has no product and who is about choosing a product from nest \( j \), conditional on buying. That is, I do not examine the optimal stopping problem.
for the moment, I assume that the decision of buying in the current period is already made and ask, which product is the optimal choice. Remember that the index \( j \) of a product identifies unambiguously its nest \( g \). This implies that we can replace \( j \) by \( g \) in the transition probability function of observed customer specific state \( y \).\(^2\) So, we can write this function as \( l(y' \mid y, g) \). Hence, expected discounted value of choosing product \( j \) belonging to nest \( g \), assuming that all future decisions will be made optimally, is:

\[
x_j \gamma_g + \varepsilon_g + (1 - \sigma_g)\varepsilon_j + \beta \int V(s')p(ds' \mid s, g).
\]

Note that the last term in this expression depends only on \( g \) and not \( j \), that is, it is the same for all products belonging to the same nest \( g \). Denote from this last term \( w_g(s) \equiv \int V(s')p(ds' \mid s, g) \), the expected value of the next period problem conditional on current choice \( g \) and state \( s \).

The probability of choosing product \( j \) belonging to nest \( g \), denoted by \( p_j \), is the conditional probability of choosing \( j \) from \( g \) times the probability of choosing \( g \). That is, \( p_j = p_{j \mid g}p_g \), or, given the nested logit distributional assumptions:

\[
p_j = \frac{\exp[(x_j \gamma_g + \beta w_g(s))/(1 - \sigma_g)]}{\mathcal{R}_g} \cdot \frac{\exp[(1 - \sigma_g)\ln R_g]}{\sum_{g' = 1}^{G} \exp[(1 - \sigma_{g'})\ln \mathcal{R}_{g'}]},
\]

where \( \mathcal{R}_g \equiv \sum_{j \in g} \exp[(x_j \gamma_g + \beta w_g(s))/(1 - \sigma_g)] \). Note that

\[
\mathcal{R}_g = \exp[\beta w_g(s)/(1 - \sigma_g)] \sum_{j \in g} \exp[x_j \gamma_g/(1 - \sigma_g)] \equiv \exp[\beta w_g(s)/(1 - \sigma_g)]R_g, \quad \text{and}
\]

\[
(1 - \sigma_g)\ln \mathcal{R}_g = (1 - \sigma_g)\ln R_g + \beta w_g(s),
\]

where \( R_g \equiv \sum_{j \in g} \exp[x_j \gamma_g/(1 - \sigma_g)] \).

As a result, we can write

\[
p_j = \frac{\exp[x_j \gamma_g/(1 - \sigma_g)]}{R_g} \cdot \frac{\exp[(1 - \sigma_g)\ln R_g + \beta w_g(s)]}{\sum_{g' = 1}^{G} \exp[(1 - \sigma_{g'})\ln \mathcal{R}_{g'} + \beta w_{g'}(s)]}.
\]

Intuitively, the first term of this formula says that choosing an alternative conditional on choice of nest \( g \) is described by a standard logit model. Here, mean utility of alternative \( j \) is \( x_j \gamma_g/(1 - \sigma_g) \). The inclusive value \( r_g \equiv \ln R_g \) is the expected maximum utility of this conditional choice problem. The second term says that choosing between nests is described by a logit model too. Now, mean utility of alternative \( g \) is given by the weighted sum of the inclusive value \( r_g \) of this nest and the discounted value of the next period problem, given current choice \( g \). Similarly, we can handle

\(^2\)From the current choice part, \( y' \) is only affected by the fact whether a purchase happened in current period \( (g \neq 0) \) or not \( (g = 0) \).
the conditional choice problem of customers who already have a product and want to upgrade it. Denote the corresponding inclusive values by \( r^u_g \).

Now we are ready to make the necessary additional assumption on transition probabilities, which will reduce the size of the dynamic problem. I assume that the customer does not form expectations about future characteristics of each individual product. Instead, she only predicts levels of expected future maximum utilities, that is, inclusive values, of each nest. In technical terms, I assume that inclusive values \( r_g \) and \( r^u_g \) are sufficient statistics to predict future values of characteristics \( x \) of products belonging to the corresponding nests. This enables me to formulate the optimal stopping problem, joint with that of choice between nests, as a dynamic programming model where component \( x \) in the vector of state variables is replaced by the inclusive values. Hence, state and choice spaces are reduced considerably. The formulation is the following.

The customer still maximizes her lifetime utility by making a choice each period from a finite set of alternatives. The choice set is \( \{0, 1, ..., G\} \), that is, inaction or one of the \( G \) nests. The observed customer specific state variable \( y \) is the same as before. Again, to specify instantaneous utility functions I consider four distinct cases according to observed customer specific states and choices:

Case 1. \( y = 0, g = 0 \).

\[
\begin{align*}
  u_g &= c + \varepsilon_0.
\end{align*}
\]

Case 2. \( y = 0, g \in \{1, ..., G\} \).

\[
\begin{align*}
  u_g &= (1 - \sigma_g)r_g + \varepsilon_g.
\end{align*}
\]

Case 3. \( y \in \{1, ..., G\}, g = 0 \).

\[
\begin{align*}
  u_g &= c_y + \varepsilon^u_0.
\end{align*}
\]

Case 4. \( y \in \{1, ..., G\}, g \in \{1, ..., G\}, g = y \).

\[
\begin{align*}
  u_g &= (1 - \sigma^u_g)r^u_g + \varepsilon^u_g.
\end{align*}
\]

The logic and assumptions are the same as before except that now the customer chooses only between nests and inaction. Within nest choices are 'integrated out': these problems are considered as already solved at this stage and their only relevant consequence being represented by inclusive values \( r_g \) and \( r^u_g \).

The state vector of this reduced problem is \( z = (r, y, \varepsilon) \) with transition probability

\[
\begin{align*}
  p(r', y', \varepsilon' \mid r, y, \varepsilon, g) &= h(\varepsilon' \mid r', y')f(r' \mid r)l(y' \mid y, g).
\end{align*}
\]
Here \( r \) is the vector of \( r_g \)'s and \( r^u_g \)'s. The corresponding Bellman equation is:

\[
V(z) = \max_{g \in \{0, 1, \ldots, G\}} \left[ u_g(z) + \beta \int V(z') p(dz' | z, g) \right].
\]

One can already specify a computationally feasible econometric model from this structure. I do so after first describing the industry and discussing how I use the data.

4 Industry, data and econometric specification

4.1 Industry

Servers are important building blocks of computer networks. Most large organizations, such as private companies, government agencies, universities, build up their own computer networks, which interconnect different computer machines. The main task of these networks is to enable employees to communicate and access to various informations and services. A typical network consists of client and server computers. Client computers are most often PCs that, in particular, allow users to access servers of the network, besides storing data and running software applications. Servers connect clients and provide network services, which can vary from network to network. Servers can provide file and print sharing; security services, such as authentication, user administration; or internet firewall protection. Also, servers can give access to databases stored on their hard disc. Software applications can be run on them or their computing power can be provided directly to client computers. Large servers can be specialized on mission critical tasks: Running specialized software on huge databases, for instance, in financial and banking areas, providing services to thousands of clients, ATM machines or dumb terminals. Smaller, so called work group servers provide print and file sharing, user administration and authentication to a smaller number (at most a few dozen) of client PCs.

Customer heterogeneity results in strong differentiation of server products. For instance, they can be different by their operating systems; by sizes of short run and hard disc memories; by the brand, number and potential number of CPUs; by CPU and mother board architecture. Differentiation shows up in production too. There are three main business models of server manufacturers: the integrated, non-integrated and open source models. The integrated approach means that the manufacturer provides both, the hardware and the operating system. This is the case for many UNIX provider (Sun, HP, e.g.), for instance. In the non-integrated approach, the hardware and the operating system provider are separated. The most important example is the Intel/Windows pair. The open source business model uses non integrated hardware platform (Intel architecture, very often) but the operating system is not copyrighted. Its source code can be freely downloaded from the Internet. The most well known open source OS is Linux. It is
interesting that a producer can use several business models at the same time. For instance, IBM sells its proprietary OS/hardware bundle, but also produces servers running Linux. Given all these differentiating factors, it is natural to think that not all servers are competing on the same market.

Indeed, servers are subjects of a recent policy decision. The European Commission has stated that Microsoft uses its quasi monopoly on the PC operating system market to control interoperability on the market for low-end servers’ operating systems to build up a dominant position in this market (see EU Commission (2004)). Using informal methods on a large customer survey, the EC has argued that relevant market for these low-end operating system products does not include those of larger systems and punished Microsoft with a record monetary fine. Using a large world-wide server dataset, Ivaldi and Lőrincz (2005) estimate a static equilibrium model and run SSNIP and full equilibrium relevant market tests. Although they do not address the Microsoft issue, that is, the relevant market of operating systems, they find some evidence that there are several relevant markets for low-end servers. The present paper builds on this result and estimates the dynamic model for low-end server products.

4.2 Data

I have a large world-wide dataset on the market level. It is detailed in Appendix A. There are observations on quantities, prices and technical characteristics for basically all server models in three regions (after some aggregation): Western Europe, Japan and the US, from the period Q1 1996 until Q1 2001. The prices are real prices denominated in Q1 1996 US dollars. I study the low-end segment of server products, which can be defined as products priced below $4000.3 The main technical characteristics are operating systems (Linux, Novell’s NetWare and Microsoft Windows NT); CPU type; CPU architecture; number of CPUs; CPU capacity; number of racks.

There are observations separately on initial server shipments (ISS), that is, sales of new servers, and upgrades. Total number of observations is 3444, 1676 and 3637 for Europe, Japan and the US, respectively.

4.3 Calibration

The number of customers already having a product at the beginning of a given period is not observed in the data. I solve this problem by calibrating some parameters. For a given country, I specify an initial installed base $ib_0$, which gives the total amount of server products, old and new, being already installed at the end of period right before the first period of the data, Q1 1996.

3Some evidence that these products constitute a relevant market can be found in Ivaldi and Lőrincz (2005).
From $ib_0$, I specify operating system specific initial installed bases for the same time. I assume that total initial installed base is split up between different operating systems proportional to their total quantities sold in the whole time interval of data for this given country. Having set operation system specific initial installed bases, for a given value of $q$, the probability of product break-down, one can calculate next period installed bases using the data of quantities sold per operating system in the current period:

$$ib_{g,t} = (1 - q)ib_{g,t-1} + q_{g,t},$$

where $ib_{g,t}$ is installed base of operating system $g$ at the end of period $t$, and $q_{g,t}$ is the quantity sold of servers with operating system $g$ in period $t$. The result is a data set, which tells the number of customers being in state $y_t = g$ ($g = 0, 1, ..., G$) in period $t$. I will use this data set in the third step of my sequential estimation procedure, see below.

Note that a value of $q$ is needed for this calibration exercise. This parameter is part of the transition probability. Estimation of the class of structural dynamic economic models, which the present one belongs to as well, assumes very often that transition probabilities are estimated separately, see Rust (1994). This is the case for Melnikov (2000), Aguirregabiria (2002) and Hendel and Nevo (2003), for example. This is needed to be able to carry out an otherwise computationally intractable estimation algorithm. So, even if $q$ did not play any role in calibration its estimation would cause serious problems. That is why its value is calibrated and, together with $ib_0$, I make a sensitivity analysis checking robustness of results for their different values.

4.4 Econometric specification

I estimate the model by extending the sequential estimation procedure of static nested logit models, proposed by McFadden (1981), to a dynamic setting. This means that first I estimate static conditional logit models of within nest choices. In the second step, I estimate transition probabilities of these models’ inclusive values. Finally, a dynamic logit model of choice between nests is estimated structurally. Here, choice specific instantaneous utilities are functions of inclusive values from the first step, and transition probabilities come from the second step. This sequential estimation procedure is consistent with the structure of economic model in Subsection 3.2.

4.4.1 Estimating inclusive values of nests

From the first part on the right hand side of (1), the log probability of choosing product $j$ conditional on choosing nest $g$ is

$$\ln p_{j|g} = x_j \gamma_g / (1 - \sigma_g) - r_g.$$
Remember that the product specific vector $x_j$ contains $K$ observed-by-the-econometrician components, denoted by $\vec{\pi}_j$, and one unobserved variable $\xi_j$. This latter is a product specific quality measure which represents information not available from the data but perceived by the customer. Denote the sample counterpart of $p_{j|g}$ by $\pi_{j|g} = q_j / \sum_{j' \in g} q_{j'}$, where $q_j$ is the quantity sold of product $j$. So, having a panel of observations on member products of nest $g$, the estimating equation is

$$\ln \pi_{j|g} = \vec{\pi}_{j,t}^g \gamma_g / (1 - \sigma_g) - r_{g,t} + \xi_{j,t} / (1 - \sigma_g).$$

The econometric error term is $\xi_{j,t} / (1 - \sigma_g)$. Since one of the components in $\vec{\pi}_{j,t}$ is price one must use two-stage least squares estimation to avoid endogeneity bias. Instruments are basis functions of the efficient polynomial approximation of optimal instruments, as proposed by Berry, Levinsohn and Pakes (1995), assuming that elements of $\vec{\pi}$ other than price are exogenous. This is a valid assumption since the equation is conditional on stopping. Instruments used are listed in Appendix A.3.

Note that $\sigma_g$ and $\gamma_g$ are not identified separately. The former will be identified in the last step of the sequential estimation procedure. The inclusive value $r_{g,t}$ is identified as the negative of a time dummy for period $t$. Hence, the main results of the first step are $G$ time series of inclusive values, each for one nest.

### 4.4.2 Transition probabilities

Next, I estimate a Markov process for the vector of inclusive values. First, the state space is further reduced. I calculate the sums $\bar{r}_t^i \equiv \sum_{g=1}^G r_{g,t}$ and $\bar{r}_t^u \equiv \sum_{g=1}^G r_{g,t}^u$. Then I specify a first order vector autoregression of these two variables:

$$\bar{r}_t = \Phi \bar{r}_{t-1} + \epsilon,$$

where $\bar{r}_t \equiv (\bar{r}_t^i, \bar{r}_t^u)'$ and $\epsilon$ is a normally distributed error term. Having an OLS estimate of $\Phi$, it is straightforward to calculate transition probabilities $f(\bar{r}^i | \bar{r})$ for any value of $\bar{r}$.

### 4.4.3 Structural dynamic estimation

To cast the problem into a dynamic logit framework, one has to formulate the relationship between nest specific instantaneous utilities and the aggregate state variables $\bar{r}_t^i$ and $\bar{r}_t^u$. I apply a polynomial series approximation:

$$r_{g,t} = \theta_0^i + \theta_1^i \bar{r}_t^i + \theta_2^i (\bar{r}_t^i)^2 + \theta_3^i (\bar{r}_t^i)^3 + \eta_t^i,$$

$$r_{g,t}^u = \theta_0^u + \theta_1^u \bar{r}_t^u + \theta_2^u (\bar{r}_t^u)^2 + \theta_3^u (\bar{r}_t^u)^3 + \eta_t^u.$$
Using OLS estimates of \( \theta \)'s, I calculate fitted values \( \hat{r}_{g,t} \) and \( \hat{r}_{g,t}^{u} \) and use these in the utility functions. Hence, there are three observed state variables: \( \mathbf{r} \), \( \mathbf{r}^{u} \) and \( y \). The Bellman equation is the following

\[
V(\mathbf{z}) = \max_{g \in \{0, 1, \ldots, G\}} \left[ \hat{u}_g(\mathbf{z}) + \beta \int V(\mathbf{z}') p(d\mathbf{z}' | \mathbf{z}, g) \right],
\]

where \( \mathbf{z} \equiv (\mathbf{r}, y, \varepsilon) \) is the state vector. Using the data, I apply the NPL algorithm of Aguirregabiria and Mira (2002) to get maximum likelihood estimates of \( c \) and \( \{(1 - \sigma_g), (1 - \sigma_g^{u}), c_g\}_{g=1}^{G} \).

## 5 Empirical results

I present preliminary results. I estimated the model on the US data. Calibrated values are \( \beta = 0.97 \), \( q = 0.0175 \) and \( ib_0 = 600000 \). I also estimated two benchmark models to compare results. The first is a simple static nested logit model of initial server shipments (new sales). Nests are operating systems. The other benchmark model is a simple optimal stopping model, similar to that of Melnikov (2000). This is the same as the full model except that there is no upgrade problem: The customer decides on when to buy a new product. Having chosen an alternative, she keeps using it until it breaks down (with the same probability \( q \) as in the full model).

Figure 2 and 4 display inclusive value estimates of \( r_{g,t} \) and \( r_{g,t}^{u} \) from the first step for initial server shipments and upgrades, respectively. The nests correspond to Linux, Novell NetWare and Windows NT operating systems. I display values only for quarters where there are observations for all three formats. Both sets of inclusive values are normalized in such a way that Netware’s value is 1 for the first quarter of 1998. In the second step, I estimate a first order VAR of the vector of aggregated state variables: \( \mathbf{r}_t \equiv (\mathbf{r}_t, \mathbf{r}_t^{u})' \). The results are displayed in Table 1.

To carry out the third step, first I have to recover format level inclusive values from the aggregated state variables. This is done by third order polynomial regressions, whose results can be found in Table 2. Also, I display fitted values from these regressions in Figure 3 and 5 for ISS and upgrade sales, respectively. Comparing these graphs with Figure 2 and 4, one can see that the fit is quite good, dynamics of inclusive value series is recovered reasonably, both in absolute and relative terms. That is, I did not lose crucial information by the state space reducing aggregation.

Table 3 presents maximum likelihood estimates of main structural parameters from the third step. Here I used observations only from Q1 1998. Actual and predicted values of market shares \( \pi_{g,t} \equiv q_{g,t} / \sum_{g'} q_{g',t} \) are displayed in Figures 6-8. These Figures plot results from the full model of this paper, as well as from the static and simple optimal stopping models. It is apparent that the
full model is better able to capture main tendencies in the segment: The rising shares of Windows NT and Linux against the decreasing NetWare. Predicted sales patterns of the static model are rather flat. The simple optimal stopping model, however, predicts significant dynamics, which goes against actual dynamics. One can conclude that the assumption of forward looking customer behavior is validated, but the right choice of dynamic structure is crucial. Certainly, persistence effects are strong enough to drive shares.

6 Conclusions

The paper presents a dynamic model of demand for durable differentiated products when consumer preferences exhibit persistence. The root of this persistence is the fact that in many cases differentiated durables can be categorized into a few number of formats. Format specific consumer knowledge then induces persistence when the consumer considers replacing or upgrading her existing product. The model is then applied to the low-end segment of computer servers. The formats are represented by the operating systems of servers. The empirical results suggest that persistence is an important factor in explaining demand for these products.

The model describes the demand side of the market, the product price and quality generating process of the supply side is modeled as an exogenous Markov process. An extension of the framework would be the explicit modelling of producers’ dynamic competition when they face the demand built up in this paper. That would involve computation, or, at least, some characterization of the Bayesian equilibrium of this dynamic framework. The literature on empirical dynamic games, and especially the framework established by Ericsson and Pakes (1995) and Pakes and McGuire (1994), has shown success in analyzing dynamic competition (for references see, e.g., Pakes (1998)). However, in the applications of this framework the demand side is simplistic, most often static. In addition, the estimation of even these models is far more complicated than in the case of dynamic demand models (see, e.g., Berry and Pakes (2000), Berry, Ostrovsky and Pakes (2004), Pakes and McGuire (2001) and Aguirregabiria and Mira (2004), which latter is a dynamic game extension of Aguirregabiria and Mira (2002)’s single agent model estimation algorithm). Hence, the incorporation of a dynamic demand side, like that of the present paper, and a dynamic supply side into one feasible structural empirical model is a challenge for future research.
A Data

A.1 Data collection and description

I have data collected by IDC, a research firm. IDC collects server data in a quarterly/annual framework built up from three main tiers (IDC (1998)). These are vendor polling, financial modeling and end-user channel surveying. The final data base is set up from these three sources after numerous and rigorous cross-checkings. In the vendor polling phase, major vendors, channel and supplier partners are interviewed, using an electronic polling form. This takes place on a quarterly, regional, country and worldwide basis. The main informations collected are vendor, family and model data, initial server shipments (ISS) and upgrade shipments, operating system shares, pricing, CPU and configuration data.

In the next step, IDC uses detailed financial models to decompose factory revenues to the vendor, family and model level. Various publicly available financial information sources, press releases and third-party reports are used. Results are cross-checked with vendor polling data to have consistency. Finally, IDC interviews thousands of end-users on an annual basis. It surveys companies from all sizes, all industries and all geographical territories. Installed base, shipment and revenue data are cross-checked with previous results. Having finished all three steps, a further global preprogrammed cross-checking is run.

The final dataset includes quarterly observations of three countries/regions (Japan, USA, West Europe) for the period Q1 1996 - Q4 1997, and of eighteen countries (Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Japan, The Netherlands, Norway, Portugal, Sweden, Spain, Switzerland, UK and USA) for the period Q1 1998 - Q1 2001. That is, in the first part of the sample there are only aggregate data for West Europe and country level observations in the second part. In all geographic territory/quarter pairs, there are observations for the major vendors (their total number is 36), their server families and models. For each vendor/family/model slot, the following technical characteristics are observed: operating system (Linux, Windows NT, NetWare, IBM OS400 and OS390, Unix, VMS and other); CPU type (IA32, CISC, RISC); CPU architecture (UP, SMP, MPP); CPU capacity; CPU count; a dummy for whether the system is rack optimized; the number of rack slots; and a dummy for PC servers. Also, observables are the number of shipments, either initial server shipments (ISS) or upgrades; and customer revenues.

IA32 type CPUs are the Intel architecture-based 32-bit processors, including Pentium, Pentium Pro and Deschutes processors. CISC, which abbreviates complex instruction set computers, is the traditional type of processing. These computers have large instruction sets, with both simple and complex instructions of variable lengths. RISC, reduced instruction set computers, have
a processor design with smaller instructions set, with fixed-length formats. It is produced by Digital, IBM, Hewlett-Packard, Silicon Graphics and Sun Microsystems. These servers typically support Unix platform software.

There are three CPU architectures observed. UP denotes uniprocessor servers, which contain only one processor. SMP, symmetric multiprocessing, denotes the capability of the architecture to support more than one processors, symmetrically. This latter means that processors have equal access to storage devices. SMP is a generalization of the UP structure, hence SMP computers can use programs written for UP machines. MPP denotes massively parallel processing, where typically a large number of processors is used, but they are not treated symmetrically in the architecture.

The CPU count is the average number of CPUs shipped per model, at a given geographical area/quarter pair. CPU capacity is the maximum number of possible CPUs per server model. It is an integer, ranging from 1 to 128. PC servers are desktop systems designed specifically as servers. These have typically enhanced capacity and redundant hardware components, relative to ‘ordinary’ PCs, and have Intel architecture.

Customer revenue is the sum spent on a given model, at a given market. It includes the price of all shipments, peripherals attached by channel and channel margin. It is measured in current US dollars. I calculate the price paid by a customer for a given model by dividing customer revenue by the number of shipments.

A.2 Necessary transformations

In microeconomic models one needs relative prices instead of nominal ones. I choose to use real prices. These are calculated as follows. First, for a given model sold in a given country/quarter market, from the current dollar price I calculate current price denominated in the currency of the country. I use the quarterly average dollar market exchange rate. Next, I calculate real price by dividing home currency denominated price by the country’s consumer price index, which I normalize to 1 in Q1 1996. Finally, I multiply the resulting real price, at each quarter and country, by the constant, average Q1 1996 dollar/home currency exchange rate. This gives real prices denominated in a common currency: the 1996 first quarter dollar. This price is real in the sense that it measures the price of a given server system, sold in a given country, relative to the 1996 first quarter value of the CPI basket of this country. It is expressed in terms of average Q1 1996 dollar to get comparability not only within but also across countries.

In the first part of the sample, i.e., from Q1 1996 to Q4 1997, to calculate the necessary CPI index and exchange rate for Western Europe I use the fact that in the second part of the

---

4All necessary macroeconomic series were downloaded from IMF’s IFS database and from the OECD website.
sample there are country level observations in this region. For the first sample period, I calculate a weighted average of CPIs of the sixteen European countries found in the second part of the sample. Weights are proportional to average total customer spending on servers in these countries between Q1 1998 and Q1 2001. The aggregate dollar exchange rate series is calculated similarly. Note that I could have used some official series, the euro exchange rate and the euro zone or EU15 CPI, for example. These series, however, cover a slightly different set of countries that I have in the data. Moreover, official weightings are more related to differences in general consumption structures across countries. Using server expenditure based weights is more appropriate in the present application, as it tracks more closely the general price inflation hitting server product customers. That is, I take into account more efficiently the mass of information I have.

To calculate shares, one must determine the number of potential customers $N_{mt}$ at a given country/quarter pair. I follow Ivaldi and Verboven (2004) choosing first a market specific base quantity and make $N_{mt}$ proportional to it. The coefficient of proportionality is called by Ivaldi and Verboven the potential market factor. I choose the yearly average of employment level of a given country as a base quantity in a given country/quarter pair, or $E_{myt}$, where $yt$ denotes the year of quarter $t$. This assumes that computational needs of customer companies depend on the number of their employees. Most generally, computer servers are used to organize work. Although in some cases they are substitutes of human work, increasing employment surely increases the number and complexity of companies’ tasks related to organization of work. This is the underlying rationale to use the aggregate employment level as the base quantity. Then, $N_{mt} = (1 + \tau)E_{myt}$, where $\tau$ is the potential market factor, whose value, after a number of trials, I set at -0.96. Product shares are given by shipments divided by the number of potential customers.

For the second part of the sample, i.e., from Q1 1998 to Q1 2001, I aggregate observations from Western European countries into one single region. Observations belonging to the same quarter, vendor, family, server model, operating system, CPU architecture, server class (PC server or not), rack type (rack optimized or not) are aggregated into one. Shipments and employment are summed, while CPU capacity, CPU count and rack slot numbers are averaged, weighted by shipments. For each individual European country, I calculate real prices as described above. These are averaged using shipments as weights, again. So, in the final data set there are observations in three regions for the whole sample period: Japan, US and Western Europe.

A.3 Instruments

To estimate the model, I use the following instruments for a given product at a given region-quarter market. First, the exogenous characteristics of the product. Second, I use a set of polynomial basis functions of exogenous variables exploiting the three-way panel structure of
the data: I calculate instruments from the same market: the number of other products of the same producer, the sum of CPU capacities of rival producers’ products, the number of firms; also, the number of other products of the same producer and the number of firms in the same group; finally, the number of other products of the same producer in the same group and same subgroup. Instruments from the other two regional markets at the same time are: the number of other products of the same producer, the sum of CPU capacities of rivals’ products; the number of competitors producing products in the same group and the number of competitors producing products in the same group and same subgroup. Finally, I use also the number of other products of the same producer at the same regional market, in different time periods; and the number of other products in the same group and same subgroup, of the same producer at the other two regional markets, in different time periods.
B Tables

B.1 Parameter estimates

Table 1: Vector Autoregression of Aggregate State Variables

<table>
<thead>
<tr>
<th></th>
<th>$r^i$</th>
<th>$r^u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r^i(-1)$</td>
<td>0.426</td>
<td>0.111</td>
</tr>
<tr>
<td></td>
<td>(0.284)</td>
<td>(0.134)</td>
</tr>
<tr>
<td>$r^u(-1)$</td>
<td>0.413</td>
<td>0.632</td>
</tr>
<tr>
<td></td>
<td>(0.305)</td>
<td>(0.144)</td>
</tr>
<tr>
<td>constant</td>
<td>-0.357</td>
<td>1.647</td>
</tr>
<tr>
<td></td>
<td>(1.002)</td>
<td>(0.474)</td>
</tr>
<tr>
<td>adjusted R$^2$</td>
<td>0.68</td>
<td>0.88</td>
</tr>
<tr>
<td>Log likelihood</td>
<td>-2.76</td>
<td></td>
</tr>
</tbody>
</table>

Note: $r^i_t$ ≡ $\sum_{g=1}^{3} r^i_{g,t}$, $r^u_t$ ≡ $\sum_{g=1}^{3} r^u_{g,t}$; standard errors in parentheses; sample: 1998:2 to 2001:1

Table 2: Polynomial Series Approximations of Inclusive Values

<table>
<thead>
<tr>
<th></th>
<th>$r^i_{\text{Linux}}$ s.e.</th>
<th>$r^i_{\text{NetWare}}$ s.e.</th>
<th>$r^i_{NT}$ s.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>-3.651</td>
<td>10.187</td>
<td>-2.888</td>
</tr>
<tr>
<td>$r^i$</td>
<td>-2.957</td>
<td>11.658</td>
<td>3.127</td>
</tr>
<tr>
<td>$(r^i)^2$</td>
<td>2.469</td>
<td>4.283</td>
<td>-0.702</td>
</tr>
<tr>
<td>$(r^i)^3$</td>
<td>-0.377</td>
<td>0.509</td>
<td>0.084</td>
</tr>
<tr>
<td>adjusted R$^2$</td>
<td>0.93</td>
<td>0.99</td>
<td>0.97</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$r^u_{\text{Linux}}$ s.e.</th>
<th>$r^u_{\text{NetWare}}$ s.e.</th>
<th>$r^u_{NT}$ s.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>28.792</td>
<td>17.628</td>
<td>-39.393</td>
</tr>
<tr>
<td>$r^i$</td>
<td>-20.363</td>
<td>11.694</td>
<td>21.618</td>
</tr>
<tr>
<td>$(r^i)^2$</td>
<td>4.796</td>
<td>2.558</td>
<td>-3.612</td>
</tr>
<tr>
<td>$(r^i)^3$</td>
<td>-0.348</td>
<td>0.185</td>
<td>0.2</td>
</tr>
<tr>
<td>adjusted R$^2$</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Note: $r^i_t$ ≡ $\sum_{g=1}^{3} r^i_{g,t}$, $r^u_t$ ≡ $\sum_{g=1}^{3} r^u_{g,t}$; sample: 1998:2 to 2001:1
Table 3: Maximum Likelihood Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Full Model</th>
<th>Optimal Stopping Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>estimate  se</td>
<td>estimate  se</td>
</tr>
<tr>
<td>$c$</td>
<td>7.748 3.009</td>
<td>0.205 0.011</td>
</tr>
<tr>
<td>$1 - \sigma_{\text{Linux}}$</td>
<td>0.569 0.195</td>
<td>0.393 0.228</td>
</tr>
<tr>
<td>$1 - \sigma_{\text{NetWare}}$</td>
<td>0.225 0.172</td>
<td>0.212 0.087</td>
</tr>
<tr>
<td>$1 - \sigma_{NT}$</td>
<td>1.090 0.239</td>
<td>0.615 0.117</td>
</tr>
<tr>
<td>$c_{\text{Linux}}$</td>
<td>7.928 3.156</td>
<td>- -</td>
</tr>
<tr>
<td>$c_{\text{NetWare}}$</td>
<td>7.912 3.156</td>
<td>- -</td>
</tr>
<tr>
<td>$c_{NT}$</td>
<td>7.866 3.156</td>
<td>- -</td>
</tr>
<tr>
<td>$1 - \sigma_{\text{Linux}}^u$</td>
<td>0.192 1.050</td>
<td>- -</td>
</tr>
<tr>
<td>$1 - \sigma_{\text{NetWare}}^u$</td>
<td>0.147 0.924</td>
<td>- -</td>
</tr>
<tr>
<td>$1 - \sigma_{NT}^u$</td>
<td>0.230 0.706</td>
<td>- -</td>
</tr>
<tr>
<td>Log likelihood</td>
<td>-0.13</td>
<td>-0.13</td>
</tr>
</tbody>
</table>

Note: Calibrated parameters: $\beta = 0.97$; $q = 0.0175$; $ib_0 = 600000$; sample: 1998:2 to 2001:1

C Figures

Figure 1: Preference Structure
Figure 2: ISS inclusive values $r_{g,t}$, $g =$Linux, Netware, Windows NT.

Figure 3: Polynomial approximations of ISS inclusive values: $\hat{r}_{g,t}$, $g =$Linux, Netware, Windows NT.
Figure 4: Upgrade inclusive values \( r_{g,t}^u \), \( g = \text{Linux, Netware, Windows NT} \).

Figure 5: Polynomial approximations of Upgrade inclusive values: \( \hat{r}_{g,t}^u \), \( g = \text{Linux, Netware, Windows NT} \).
Figure 6: Linux actual and predicted shares $\pi_{\text{Linux},t}$ and $\hat{\pi}_{\text{Linux},t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.

Figure 7: NetWare actual and predicted shares $\pi_{\text{NetWare},t}$ and $\hat{\pi}_{\text{NetWare},t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.
Figure 8: Windows NT actual and predicted shares $\pi_{NT,t}$ and $\hat{\pi}_{NT,t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.
References


BERRY, S. T., M. OSTROVSKY, A. PAKES, [2004]: "Simple Estimators for the Parameters of Discrete Dynamic Games (with Entry/Exit Examples)," mimeo, Yale University and Harvard University.


IDC, [1998]: "Bulletin. IDC’s Server Market Methodology, Database Structure, and Definitions.”


MELNIKOV, O., [2000]: "Demand for Differentiated Durable Products: The Case of the U.S. Computer Printer Market,” mimeo, Yale University.


