Persistence Effects in a Dynamic Discrete Choice Model. Application to Low-End Computer Servers.[1]

Szabolcs Lőrincz
University of Toulouse (GREMAQ - MPSE)
szabolcs.lorincz@univ-tlse1.fr

JOB MARKET PAPER

October 7, 2005

[1] **Acknowledgements:** This paper has been written during my visit at KU Leuven, whose hospitality I acknowledge with gratefulness. I am indebted to Marc Ivaldi, my advisor at the University of Toulouse, for help and guidance, to Ariel Pakes, Manuel Trajtenberg, Johannes Van Biesebroeck and Frank Verboven for comments and insights. I thank the audiences of the 2^{nd} CEPR School in Applied Industrial Organization in Munich, the Summer Workshop of the Institute of Economics in Budapest, the 2005 Annual Conference of EARIE and seminar participants at KU Leuven and the University of Toulouse for helpful discussions. Financial help from the European Community's Research Training Network on 'Competition Policy in International Markets' and the Young Economist Award of EARIE are acknowledged. Any remaining errors in the paper are solely mine.

Abstract

The paper introduces 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 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. The model is estimated on a data set of low-end computer servers, where formats are represented by operating systems. For this application, the model can be considered as a proxy of a computer network building customer who cares not only about the likely future quality of the individual computers, but also about the direction of evolution of the network. That induces even stronger forward looking behavior than a simple optimal stopping problem. The 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 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 the preferences.

This paper proposes the introduction of format specific preference persistence into a dynamic model of demand for differentiated durable goods. It starts from recent results in the econometric modeling of Markovian discrete decision processes. Rust (1987) constructs a dynamic logit model describing the 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 the choice from a set of differentiated durable goods whose quality improves stochastically over time. The present paper couples 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.

The choice problem is the following. At the beginning of each period, the consumer knows her state: whether she has no product or has one; the quality of products that currently can be bought at the market; and her idiosyncratic preference shocks. If she has no product she can either buy one or wait and see. If she already has one she can upgrade it, which means an improvement or development, without scrapping the old product (upgrading is not compulsory). However, this upgrade is format specific and this is the root of the preference persistence. Finally, after the choice has been made there is a random draw which can result in the breakdown of the customers' existing good (if she has one at that time). In the breakdown case the customer will find herself as having no product at the beginning of the next period.

So, dynamics is generated by two sources in the model. First, the 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 the 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, the 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 (the timing of upgrading of an existing product is again an optimal stopping problem). In some sense, the possibility of future lock-in motivates even stronger forward-looking behavior than a simple optimal stopping problem.

The model is applied 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.

Servers are not simply highly differentiated durable goods but also very important parts of a computer network. A typical customer (e.g., a firm, a large institution, a university etc.) builds its information technology system continuously: The network is ever changing, evolving, growing and being upgraded; meanwhile the demand for its functionalities, size and scope can be significantly altering through time. Hence, IT managers are facing a serious planning task as they have to somehow predict the likely future evolution of their system under control. A good strategic choice can secure valuable future flexibility as well as the ability to change or specialize. On the other hand, undesired lock-in to a badly chosen system can prove to be painfully costly if later significant, and potentially unforeseen, modifications must be done. The result is that the right choice of a server requires an even more strategic and forward looking behavior than the more standard optimal timing problem of purchase of a more ordinary durable good. All these features motivate the empirical application of the model of durable goods with preference persistence.

The class of discrete decision processes, which this model belongs to, is surveyed by Rust (1994) . In particular, a dynamic nested logit model is specified where nests represent formats, that is, different operating systems. Using a world-wide panel of server purchases, the paper proposes a dynamic extension of the sequential estimation procedure of McFadden (1981): First, static conditional logit models of the nests are estimated. The second step is the transition probability matrix of the first step's nest specific inclusive values, which are the expected maximum utilities of choice from a nest. Finally, a dynamic logit model of the choice between nests is set up, where utility of a nest is a function of its inclusive value. In this step, maximum likelihood estimators are obtained using the nested pseudo likelihood (NPL) algorithm of Aguirregabiria and Mira (2002). The model is built on three main principles: product differentiation, durability (or optimal stopping), and persistence effects. The product differentiation is identified from the cross-sectional variation of the data. The optimal stopping structure is identified from the time dimension. Finally, persistence effects can be identified from the fact that the data contains not only observations of initial server shipments (i.e., sales of new servers) but also of upgrade shipments.

The model is compared to two benchmarks: 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 the 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 the related literature. Section 3 sets up the model. Section 4 discusses industry, data and econometric specification. Section 5 presents the empirical results and Section 6 concludes.

2 Related literature

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 heterogenous consumers choosing the alternative providing the highest utility. Consumer heterogeneity is mainly modeled by assuming functional forms describing the unobserved components of preferences. The static framework means that the consumers do not consider the 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 the models are more heterogenous. For example, when the 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 the optimal purchase time. In a seminal paper, Rust (1987) has set up a dynamic version of McFadden's logit model to describe and analyze empirically the problem of an agent who chooses the optimal time of replacing a used durable, which is otherwise non-differentiated (bus engines, in his application). 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, and model demand for printer machines. This is very important in a broader sense for most modern durable goods such as computers, printers, cars, etc., whose quality is rapidly improving over time. Gordon (2005) presents and estimates a dynamic demand model for PC processors. Cho (2002) builds a model that is closer to the present one. It describes the decision of the IT managers of a telecommunication firm on replacing existing mainframe computers.

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 the consumer choice of experience goods. Here, the process which links the decisions in different time periods is the consumers' learning about a new product. This process is largely affected by the producers' advertizing behavior. These effects can be analyzed by the structural model.

Network externality can be another important source of dynamics. 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 the expected future number of purchasers of each formats, already at the period of the 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 a 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 is, again, valued by consumers.

The model presented in this paper is closely related to the work of Rust and Melnikov. It adds a persistence effect to their optimal stopping model. This persistence in the preferences can be the result of learning. However, unlike Ackerberg, and mainly due to data limitations, the present paper does not model explicitly the process of learning. Another interesting issue is the possible existence of indirect network externalities in the case of server OSs. 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. Also, servers, more often than PCs, must interoperate with each other directly. Nevertheless, 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. Still, this is not to say that potential network externalities purely among servers, or their operating systems, cannot be important. The practical point is that the paper emphasizes an other crucial feature, persistence, which is likely to be more important than in the case of PCs, and which can at least be identified, given the data at hand.

3 The model

To give an overview of the model, it is useful to start with a simpler structure of Melnikov (2000) and show how the full model relates to this. First, in a simple optimal stopping problem of durable purchase the customer does not buy a good each period. In fact, if its lifetime is infinite, and assuming no multiple product ownership, she buys only one and once in her life. But independently of this, there are newer and newer products on the market each period. The dynamic optimization problem is to decide when to buy. Each period the customer decides to buy or not to buy until once she makes a purchase, and from that time on she is said to be 'out of the market', which means that from this time on she does not need to, nor she can make any purchase decision. To decide in a given period, she needs to know 1. the primitives of the model (utility parameters, discount factor and transition probabilities), and 2. the current state of the world. The state of the world consists of three elements: one is a dummy variable saying whether the customer has a product already or not; the second is a stochastic customer specific i.i.d. preference shock; and the last component denotes the level of quality of the current new products on the market and evolves stochastically as an exogenous Markov process. For the optimal decision, the customer makes a prediction of the likely future evolution of the state of the world. (Of which the dummy part evolves deterministically.)

If the decision is to buy a product the customer will keep it forever and its quality will not change. But the quality of the new products in each period still continues changing, and the state of the world too, even though, this does not affect the quality of the customer's existing product. In this simpler model of Melnikov, it is assumed that when the customer decides to stop she receives the net present value of the infinite utility flow of the product she has just chosen.

After purchase she no longer receives any payoff. Each period when the customer does not have a product and not even buys any she receives a constant 'continuation value.'

This is the simple version of Melnikov's model. Next, he makes a slight modification by assuming that even if the customer already has a product, she is still not completely out of the market because her product can break down with a given probability, and so she is facing again the dilemma of buying a new good or not. Actual breakdown reveals until each period begins. (Consequently, here already the dummy part of the state evolves stochastically, too.)

Last, the model in the present paper makes a step even further from here. It assumes all these above structures and adds one more option: Even if the customer already has a product and even if it is not broken down at the beginning of the current period she is still not out of the market: She can upgrade her existing product. Actually, this can be imagined as a change in the quality of her existing product, but it is still different from buying a completely new one as, in fact, when upgrading, the customer does not scrap her existing product. She rather modifies it, builds it larger, better, more secure, stable, adds a new feature to it etc. - complying with its format, that is, its operating system. If the customer does not upgrade then the quality of her existing product does not change. But again, the quality of the products that can be bought currently on the market is still changing - this does not depend on the customer, this is the exogenous technological progress. (The dummy part of the state is modified: it is zero in case of not having any product, else it is equal to the index of the format of the owned good.)

In this full model the payoff structure is modified a little, relative to Melnikov, to capture lock-in: Indeed, when the customer buys a completely new good she still receives a once and for all utility sum. But from that time on, in each following period, if the product does not break down, she also receives a format specific 'continuation value.' When upgrade is chosen, the customer receives a format and upgrade specific utility sum in that period. To sum, in the full model the customer not only takes into account the likely future evolution of the quality of new products when deciding about when to stop, but she also thinks of the future possibility of being locked into the specific format of the choice product, so the expected future quality levels of upgrades, complying with the predetermined format, must also be taken into account.

The next two subsections specify this decision problem as a dynamic nested logit model. First, a general model is set up 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 a situation is modeled where the customer predicts only aggregate quality processes of nests, instead of those of individual products themselves. In this simplified model the 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 a 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_t} 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: $J_t = J(s_t)$. The evolution of the state vector is governed by a Markov-transition probability, which is assumed to be known by the customer: $p(s_{t+1} \mid s_t, j)$.

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

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

Here, s' denotes the next period state vector. 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 the current period, and $y = g$ means that she already has a product at the beginning of the current period and this product belongs to nest g. Finally, ε represents customer and product specific heterogeneity, which is unobserved by the researcher.

Now we are ready to set up the instantaneous indirect utility functions. There are four distinct cases according to the 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. $y \in \{1, ..., G\}$, $g = 0$, $(j = g)$.

$$u_j = c_y + \varepsilon_0^u.$$

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

$$u_j = x_j^u\gamma_g^u + \varepsilon_g^u + (1 - \sigma_g^u)\varepsilon_j^u.$$

Figure 1 displays graphically this preference structure: In Case 1, the customer does not own any product at the beginning of the current period and she does not buy anything either. Her payoff is a constant c, and her specific valuation ε_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 $x_j\gamma_g$, where γ_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 the payoffs in a situation where the customer already has a product and decides to 'upgrade' it. 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 the empirical application, nests corresponds to different operating systems. So, it is assumed 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 the distributions of unobserved heterogeneity terms ε, see, e.g., Cardell (1997): ε_0 and ε_0^u are distributed identically and independently across all alternatives and periods with extreme value distributions. The terms $\varepsilon_g + (1 - \sigma_g)\varepsilon_j$ and ε_g are distributed identically and independently across nests and periods with extreme value distributions. The same is true for $\varepsilon_g^u + (1 - \sigma_g^u)\varepsilon_j^u$ and ε_g^u. $\sigma_g \in (0, 1)$ governs within group correlation in nest g. If its value converges towards 1 then for $j \in g$ the correlation between the ε_j terms converges towards 1. If σ_g goes to zero then the same correlation goes to zero too.

As a last step, transition probabilities are specified. A form of conditional independence is assumed:

$$p(x', y', \varepsilon' \mid x, y, \varepsilon, j) = h(\varepsilon' \mid x', y')f(x' \mid x)l(y' \mid y, j).$$

First, the next period values of the unobserved heterogeneity terms ε 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, the product specific characteristics in x evolve as an exogenous Markov-process. In particular, customer choices do not affect this quality generating process. Third, the 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 $l(y' \mid y, j)$:

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

A customer who in the current period does not have any product ($y = 0$) and does not buy a new one either, will find herself in this same state at the beginning of the 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. 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 $g = y$.

Note that a customer can have at most one product (upgraded or not) at a given period. Since the data are on the market level, multiple purchases and multiple ownership are not observed. Hence, the break-down probability q could also be interpreted as a probability with which a customer, who already has a product, has a need of buying a new computer. Also, one can think of the case of break-down as having a new IT manager or having a need for a completely new system functionality. However, there is no sufficient information in the data to distinguish these cases from that of the breakdowns.

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

The state and choice spaces are reduced 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, the instantaneous utilities of nests are represented by functions of the inclusive values from static conditional logit models describing *within* nest choice of products. The model works as follows.

Consider the case of a customer who has no product and who is about choosing a product from nest g, conditional on buying. That is, we do not examine the optimal stopping problem for the moment, and 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 the observed customer specific state y.[1] So, this function can be written as $l(y' \mid y, g)$. Hence, the 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. Intuitively, the next period customer specific observed state y' does not depend on the current choice j but only on its format g: If the current choice j belongs to format g than $y' = g$ (unless a product breakdown happens, when $y' = 0$, regardless of g). In short, the customer specific persistence is carried through time by the format of the product and not the product itself. Denote $w_g(s) \equiv \int V(s')p(ds' \mid s, g)$, the expected value of the next period problem conditional on the 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, $p_j = p_{j|g}p_g$, or, given the nested logit distributional assumptions:

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

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

$$\begin{aligned} \overline{R}_g &= \exp[\beta w_g(s)/(1-\sigma_g)]\sum_{j\in g}\exp[x_j\gamma_g/(1-\sigma_g)] \equiv \exp[\beta w_g(s)/(1-\sigma_g)]R_g, \text{ and} \\ (1-\sigma_g)\ln \overline{R}_g &= (1-\sigma_g)\ln R_g + \beta w_g(s), \end{aligned}$$

[1] From the current choice part, y' is only affected by the fact whether a purchase happened in the current period ($g \neq 0$) or not ($g = 0$).

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

As a result, the following can be written:

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

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, the 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, the 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, one can handle the conditional choice problem of customers who already have a product and want to upgrade it. Denote the corresponding inclusive values by r_g^u.

Now we are ready to make the necessary additional assumption on transition probabilities, which will reduce the size of the dynamic problem. The crucial assumption introduced at this stage is that the customer does not form expectations about future characteristics of each individual product. Instead, she only predicts the levels of expected future maximum utilities, that is, inclusive values, of each nest. In technical terms, the assumption states that the inclusive values r_g and r_g^u are sufficient statistics to predict the future values of characteristics x of products belonging to the corresponding nests. This makes possible to formulate the optimal stopping problem, joint with that of the choice between nests, as a dynamic programming model where component x in the vector of state variables is replaced by the inclusive values. Hence, the state and choice spaces are reduced considerably.[2] 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 the instantaneous utility functions consider the four distinct cases according to the observed customer specific states and choices:

[2]McFadden (1981) proposes sequential estimation of static nested logit models by first estimating simple static conditional logit models of the within nest choices, then to estimate a simple static logit model of the choice between nests. In this latter, the 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 the inclusive values of product groups as sufficient statistics for the price processes of these groups. 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.

Case 1. $y = 0$, $g = 0$.

$$u_g = c + \varepsilon_0.$$

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

$$u_g = (1 - \sigma_g) r_g + \varepsilon_g.$$

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

$$u_g = c_y + \varepsilon_0^u.$$

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

$$u_g = (1 - \sigma_g^u) r_g^u + \varepsilon_g^u.$$

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 the inclusive values r_g and r_g^u.

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

$$p(r', y', \varepsilon' \mid r, y, \varepsilon, g) = h(\varepsilon' \mid r', y') f(r' \mid r) l(y' \mid y, g).^3$$

Here r is the vector of r_g's and r_g^u's. The corresponding Bellman equation is:

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

One can already specify a computationally feasible econometric model from this structure. The next section do so after first describing the industry and discussing how the data are used.

4 Industry, data and econometric specification

4.1 Industry

Servers are important building blocks of computer networks.[4] 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 the

[4] For more economic and technical analysis of the server industry, see EU Commission (2004), Ivaldi and Lőrincz (2005) and Van Reenen (2003).

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 the 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 the client computers. Large servers can be specialized on mission critical tasks: Running specialized software on huge databases, for instance, in the 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. Microsoft reportedly argued that the market was larger. 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

To estimate the model, a large world-wide market level dataset is used. 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. The model is used to study the low-end segment of server products, which can be defined as products priced below \$4000.[5] 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. Region level total numbers of observations are 3413, 1666 and 3605 for Europe, Japan and the US, respectively. So, the total number of observations is 8684.

4.3 Calibration

The number of customers already having a product at the beginning of a given period is not observed in the data. This problem is solved by calibrating some parameters. For a given country, an initial installed base ib_0 is specified, which gives the total amount of server products, old and new, being already installed at the end of the period right before the first period of the data, Q1 1996. From ib_0, and for the same time period, operating system specific initial installed bases are specified. It is assumed that total initial installed base is split up between different operating systems proportional to their total quantities sold in the whole time interval of the 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 the 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. This data set will be used in the third step of the sequential estimation procedure, see below.

[5]Some evidence that these products constitute a relevant market can be found in Ivaldi and Lőrincz (2005).

Note that a value of q is needed for this calibration exercise. This parameter is part of the transition probability function. 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 the calibration its estimation would cause serious problems. That is why its value is calibrated and, together with ib_0, a sensitivity analysis is carried out checking robustness of the results for their different values. Given micro level observations, it would certainly be interesting to estimate q, and especially to specify different values for different formats.

4.4 Econometric specification

The model is estimated by a sequential estimation procedure: First, static conditional logit models of within nest choices are specified. In the second step, the transition probabilities for these models' inclusive values are estimated. Finally, using the results from the first two steps a dynamic logit model of choice between nests is estimated structurally. This sequential estimation procedure is consistent with the structure of the economic model in Subsection 3.2.

4.4.1 Inclusive values

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 $\overline{x}_j$, and one unobserved variable ξ_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 $\pi_{j|g} \equiv q_j / \sum_{j' \in g} q_{j'}$, 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

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

The econometric error term is $\xi_{j,t}/(1 - \sigma_g)$. Since one of the components in $\overline{x}_{j,t}$ is price one must use two-stage least squares estimation to avoid endogeneity bias. Instruments are basis functions of the efficient polynomial approximation of the optimal instruments, as proposed by

Berry, Levinsohn and Pakes (1995), assuming that elements of $\overline{x}$ other than price are exogenous. This is a valid assumption since the equation is conditional on stopping.

Note that σ_g and γ_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, a Markov process for the vector of inclusive values is estimated. First, the state space is further reduced. The following sums are calculated: $\overline{r}_t^i \equiv \sum_{g=1}^G r_{g,t}$ and $\overline{r}_t^u \equiv \sum_{g=1}^G r_{g,t}^u$. Then a first order vector autoregression is specified for these two variables:

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

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

4.4.3 Structural dynamic estimation

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

$$\begin{aligned} r_{g,t} &= \theta_0^i + \theta_1^i \overline{r}_t^i + \theta_2^i \left(\overline{r}_t^i\right)^2 + \theta_3^i \left(\overline{r}_t^i\right)^3 + \eta_t^i, \\ r_{g,t}^u &= \theta_0^u + \theta_1^u \overline{r}_t^u + \theta_2^u \left(\overline{r}_t^u\right)^2 + \theta_3^u \left(\overline{r}_t^u\right)^3 + \eta_t^u. \end{aligned}$$

Using OLS estimates of θ's, the fitted values $\widehat{r}_{g,t}$ and $\widehat{r}_{g,t}^u$ can be calculated and used in the utility functions. Hence, there are three observed state variables: $\overline{r}^i$, $\overline{r}^u$ and y. The Bellman equation is the following

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

where $\widetilde{z} \equiv (\overline{r}, y, \varepsilon)$ is the state vector. Using the data, the nested pseudo likelihood (NPL) algorithm of Aguirregabiria and Mira (2002)[6] is applied to get maximum likelihood estimates of c and $\left\{(1-\sigma_g), \left(1-\sigma_g^u\right), c_g\right\}_{g=1}^G$.[7]

[6] The programs were downloaded from the website of Victor Aguirregabiria. Some parts have been modified, or replaced with newly written codes.

[7] 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.

5 Empirical results

A baseline model is estimated using calibrated values of $\beta = 0.97$, $q = 0.0175$ and $ib_0 = 600000$. To compare the results, two benchmark models are also estimated. The first is a simple static nested logit model of initial server shipments (new sales). Nests are operating systems. This model was obtained simply by setting $\beta = 0$. 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).

5.1 Inclusive values

Tables 1 and 2 presents conditional logit estimates from the first step. Here there are five estimated equations for initial server shipments (ISS), each for an operating system as a nest, and five for the upgrades. The operating systems are Linux, Novell NetWare, Windows NT, Unix and other OSs. Each pair of columns in the tables represent a regression with standard errors. The price regressors were treated as endogenous and the estimations were carried out by two-stage least squares method. The instruments used are listed in Appendix A.3. The price coefficients are negative, as expected. The reported first stage R^2s indicate that the instruments provide reasonable approximations of the endogenous regressors. Other regressors include a set of producer specific fixed effects, characteristics and time dummies. In most cases, rack optimization (rack), number of racks, number of CPUs (ccount) and possible number of CPUs (ccapac) are features that are valued positively by the customers. PC servers are server computer which are basically enhanced PCs and they are relatively less valued. That might reflect the fact that these are simpler machines, which can have relatively lower quality.

Figure 2 and 4 display inclusive value estimates of $r_{g,t}$ and $r^u_{g,t}$ from the first step for initial server shipments and upgrades, respectively. These are the normalized values of the negatives of time dummies from Tables 1 and 2. Both sets of inclusive values are normalized in such a way

In an outer circle, it searches the parameter vector which maximizes the log likelihood. Alternative estimation methods to tackle the curse of dimensionality include, for example, the randomization by Rust (1997), or the simple nonparametric conditional choice probabilities (CCP) estimator of Hotz and Miller (1993). The NPL '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 the 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, the computational burden can be much smaller.

that Netware's value is 1 for the first quarter of 1998. Values are displayed only for quarters where there are observations for all five formats. Since the inclusive value of a choice set is the expected maximum utility from the set, the time series can be interpreted as quality processes of the different formats. As for initial server shipments, there is a strong tendency of Linux's valuation increasing more than those of others. Windows NT and Novell NetWare are valued similarly and increasing moderately. For the upgrade quality processes, one can see a bigger difference between Windows and NetWare. Also, Linux's dynamics is more similar to those of the other formats and the Unix format displays a larger growth than for initial server shipments. These differences in relative dynamics provide the basis of identification of the full model relative to the simple optimal stopping model.

5.2 Transition probabilities

The second step, is the estimation of a first order VAR of the vector of aggregated state variables: $\overline{r}_t \equiv (\overline{r}^i_t, \overline{r}^u_t)'$. The two components of this vector are sums of inclusive values from the first step for ISS and upgrades, respectively. The results are displayed in Table 3. Both equations provide a relatively good fit, despite the small sample. Own lagged variables seem more important in explaining future values in both cases. In principle one could specify a richer dynamic structure for the transitions, for example, by a higher order VAR. However, the time span is relatively short making the estimation less efficient. Also, the state space would be larger, posing computational problems for the next step. Finally, an argument in favor of the first order VAR is that the correlograms of the residual series shows no significant autocorrelation at any lag.

5.3 Structural dynamic estimation

To carry out the third step, first the format level inclusive values have to be recovered from the aggregated state variables. This is done by third order polynomial regressions, whose results can be found in Table 4. Each pair of columns represent a regression of the respective formats inclusive value on first, second and third powers of the aggregated state variables $\overline{r}^i_t$ and $\overline{r}^u_t$, separately for ISS and upgrades processes. Note that the goal of this exercise is to have the best possible in-sample fits. First, this can be measured by the R^2s of the regressions. These statistics are impressively high, perhaps a little weaker for the Other format, but still strong in that case too. Second, fitted values from these regressions are displayed 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, the state space reducing aggregation does not lead to loss of crucial information.

Table 5 presents maximum likelihood estimates of main structural parameters from the third step. Here observations only from Q1 1998 are used to have a sample with each format observed in each period. The last two columns of this table present the coefficients and standard errors for the full model. The myopic and simple optimal stopping results are also displayed. Note that in this third step pretty complicated functions of the estimated first step coefficients are used as explanatory variables. One should take this fact into account when calculating standard errors of parameters. Instead of generalizing the correction method of the estimated variance-covariance matrix from sequential estimation of static nested logit models by McFadden (1981), a bootstrap is carried out to get standard errors. In one bootstrap round first the original, product level dataset is resampled by generating two random integer index vectors (for ISS and upgrades) with replacement, with sizes equal to the numbers of observations in the dataset. The elements of the bootstrap sample became the observations with index equal to the random vectors elements. Then all the three steps of the above outlined estimation procedure are repeated for this sample. Resampling rounds were carried out 1000 times, independently. This process generated an empirical distribution for the third step parameter estimates.

The estimated σ parameters do not seem to be particularly stable across models. That can be a result of mispsecification of some (or all) of the structures, and also of the small sample size (note that in the third step only the time series are being used from the previous steps). Hence, one must be cautious to draw conclusions. In the full model the Windows initial shipments have the highest complement of the σ parameter, and NetWare and Other OSs the lowest. This is consistent with the data, where Windows ISS shares are increasing over time and NetWare is decreasing (see below). The upgrade parameters seem to show no pattern that is similar to that of ISS estimates. The constant and continuation terms are more precisely estimated than the others, though they are less dispersed.

Tables 6-8 present some analysis of sensitivity of the parameter estimates for the full model. It seems that the results are not too sensitive to the choice of the discount factor's value (in the reasonable range). Increasing the breakdown probability decreases the likelihood of the sample, showing that servers are not especially often replaced. It would have been interesting to estimate this parameter, and especially by specifying different values to different formats, but that would have amounted to asking too much from the data. The results are relatively not sensitive to the choice of initial installed base value. Here the likelihood increases with an increase in the calibrated parameter value. Overall, the results might indicate potential identification problems with these parameters, coming from the rather restricted nature of the data (short time period, market level observation, no information on usage etc.), which validates calibration as opposed to estimation.

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, as well as from the static and simple optimal stopping models. Neither of the models provide a particularly tight fit. This is not surprising given the rather limited time span of the data. Nevertheless, there are significant differences between the explanatory powers of the models. The static and optimal stopping models performs poorly in capturing stylized dynamic features in the segment, as, for example, the rising shares of Linux and Windows and the declining patterns of NetWare and Unix. The static model display flat shares for Unix, NetWare and Other OSs, but shows surprising, and mostly incorrect, dynamics for the rest. The simple optimal stopping model is even worse, especially in capturing the levels of shares. On the contrary, the full model captures mostly correctly both the levels and dynamics of shares. It provides no apparent false fit in any case. In particular, it predicts rising shares of Windows NT and Linux against the decreasing NetWare, though the results are still rather rough.

Why is this difference of explanatory power across models? First, one can conclude that the assumption of forward looking customer behavior is validated. The choice of a durable good, given a technology improving unstoppably, certainly creates an optimal stopping problem. In this sense, the static model is ill specified as it assumes that customers do not think about the future and make decisions at each period as if this had no consequences for later times. The poor fit of the optimal stopping model is more surprising. This result highlights that the right specification of the dynamic structure is crucial. Servers bought today must fit optimally in an ever changing network and the choice of the systems' format is a very important strategic step. The full model proxies this strategic aspect of the behavior of the customers in this industry by modeling the upgrade decisions. These latter induce persistence effects that seem strong enough to drive shares. Customers do take into account the expected future levels of upgrade qualities when making a purchase decision of a new server computer. Even though actual networks, and their evolution, are not observed in this aggregate data their effect can be captured indirectly in the market level sales dynamics. This is the main contribution of the model in this paper.

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 the demand for these products.

Since the model is written to an aggregate data, it can be thought of as only a proxy describing the behavior of a network building customer. This underlying building process requires stronger planning and forecasting ability from the decision maker than a simple timing problem of a durable good purchase: It is not only the dimension of the individual product quality improvement over time that is of importance, but also an other one, which captures the overall quality of the network, and which shows a dynamic pattern, too. This rather micro level phenomenon is then translated into the aggregate upgrade process and hence provides the basis of identification of the model.

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 the 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 (2000)). However, in the applications of this framework the demand side is simplistic, most often static.[8] In addition, the estimation of even these models is far more complicated then in the case of the dynamic demand models (see, e.g., the method of Aguirregabiria and Mira (2004), which is a dynamic game extension of the single agent model estimation algorithm of Aguirregabiria and Mira (2002), also the algorithms of Bajari, Benkard and Levin (2005), Berry, Ostrovsky and Pakes (2004), Berry and Pakes (2000) and Pesendorfer and Schmidt-Dengler (2004)). 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.

[8] An early attempt to incorporate dynamic demand, though with some strong restrictions on the dynamic competition, is provided by Carranza (2005). Park (2004) presents a full dynamic model, but its solution is not explicit, and the estimation is semi-parametric.

A Data

A.1 Data collection and description

The data is 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. The price paid by a customer for a given model is calculated by dividing customer revenue by the number of shipments.

A.2 Necessary transformations

In microeconomic models one needs relative prices instead of nominal ones. The paper uses real prices. These are calculated as follows. First, for a given model sold in a given country/quarter market, from the current dollar price current price denominated in the currency of the country is calculated. Here the quarterly average dollar market exchange rate is used.[9] Next, real price is calculated by dividing home currency denominated price by the country's consumer price index, which is normalized to 1 in Q1 1996. Finally, the resulting real price is multiplied, 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 the fact can be exploited that in the second part

[9] All necessary macroeconomic series were downloaded from IMF's IFS database and from the OECD website.

of the sample there are country level observations in this region. For the first sample period, a weighted average of CPIs of the sixteen European countries, found in the second part of the sample, is calculated. 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 some official series, the euro exchange rate and the euro zone or EU15 CPI, for example, could have been used. These series, however, cover a slightly different set of countries that are 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, the mass of information in the data is more efficiently taken into account.

To calculate shares, one must determine the number of potential customers N_{mt} at a given country/quarter pair. The paper follows 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. The base quantity in a given country/quarter pair is the yearly average of employment level of a given country, 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 τ is the potential market factor, whose value, after a number of trials, its value is 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, observations from Western European countries are aggregated 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, real prices are calculated 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.

The range of price variable goes from a few thousands to several millions of dollars. Obviously, not all of these products are on the same market. However, given the lack of some crucial information (e.g., functionalities of servers), the definition of the relevant markets can only be approximate, at best. Ivaldi and Lőrincz (2005) provide results from econometric implementation of relevant market tests using the same data and a static model of industry equilibrium. Based on these calculations, the analysis is restricted to servers priced below $4000. This segment vaguely corresponds to smaller workgroup servers that provide printing, file sharing, user authentication and some other communication services to client computers. The total number of observations is 8684.

A.3 Instruments

To estimate the five equations in the first step of the sequential procedure, the following instruments are used for a given product at a given quarter. First, in each equation, the exogenous characteristics of the product. Second, a set of polynomial basis functions of exogenous variables is used exploiting the panel structure of the data: In each equation, the sum of CPU capacities of the producers' own products, the sum of CPU capacities of rival producers' products, the sum of CPUs of the producers' own products and the sum of CPUs of rival producers' products. In addition, the number of own products and the number of competitor's products of a given producer are used for NetWare, Windows, Unix and Other operating systems. For Linux, NetWare, Unix and Other OS equations the number of firms is used also.

B Tables

B.1 Parameter estimates

Table 1: Conditional Logit Estimates for Initial Server Shipments

parameter	Linux	s.e.	NetWare	s.e.	NT	s.e.	Unix	s.e.	Other	s.e.
price	-475.00	322.93	-807.08	292.31	-334.13	318.79	-408.01	155.00	-113.78	89.69
1^{st} stage R^2		0.77		0.79		0.77		0.88		0.82
firm dummies										
Acer	-1.56	1.01	1.46	1.14	0.07	1.25	-1.12	0.27	-1.17	0.44
AST			-1.99	0.63	-2.10	0.62	-3.92	0.66	-3.18	0.71
Compaq	-1.23	0.26	0.56	0.24	0.83	0.22	0.12	0.23	0.11	0.27
Dell	-0.50	0.39	0.47	0.26	1.11	0.31	0.02	0.26	0.07	0.32
Digital			-0.90	0.56	0.24	0.56	-0.72	0.62	-0.90	0.63
Fuji-Siemens	-2.58	0.51	0.08	0.51	-0.08	0.49	-0.75	0.51	-2.57	0.66
Gateway	-3.18	0.51	-1.16	0.36	-0.87	0.35	-1.93	0.36	-1.28	0.38
Hewlett-Packard	-0.82	0.42	0.87	0.28	1.23	0.30	0.35	0.29	0.83	0.32
Intergraph					-3.47	1.13			-3.90	0.94
Micron			-0.69	0.84	0.10	0.83	-1.11	0.83	-1.19	0.93
Siemens	-1.52	0.93	-2.45	0.42	-2.12	0.43	-1.74	0.42	-1.33	0.43
Toshiba	-2.49	0.40	-1.45	0.36	0.10	0.47	-2.86	0.53	-2.13	0.61
Unisys	-3.12	0.95	-2.03	0.59	-1.45	0.79	-2.26	0.48	-2.92	0.58
Hitachi	-3.85	0.58	-1.58	0.48	-0.56	0.43	-0.72	0.85		
IBM	-0.22	0.39	0.66	0.31	0.99	0.29	0.21	0.27	0.63	0.31
Fuji	-1.74	0.77	0.47	0.52	0.29	0.49	-0.95	0.60	-2.36	0.64
NEC	-2.38	0.45	-0.70	0.34	-0.27	0.35	-1.55	0.32	-1.39	0.36
Olivetti			-1.05	0.54	-1.56	0.51	-2.89	0.40	-1.89	0.45
Mitsubishi	-3.20	0.99	-1.26	0.41	-0.39	0.36	-2.41	0.53	-2.15	0.53
Va Linux	-1.89	0.47								

Note: Each second column denotes a conditional logit regression where the dependent variable is $\pi_{j|g} \equiv q_j / \sum_{j' \in g} q_{j'}$, where q_j is the quantity sold of product j, and g denotes the column heading. Number of observations: 4061; 2SLS estimation; instruments listed in Appendix A.3. CISC: dummy for complex instruction set computers; rack: dummy for rack optimized servers; ccapac: potential number of CPUs; ccount: actual number of CPUs.

Table 1 (cont.): Conditional Logit Estimates for Initial Server Shipments

parameter	Linux	s.e.	NetWare	s.e.	NT	s.e.	Unix	s.e.	Other	s.e.
characteristics										
PC server	-3.42	1.91			-3.80	1.86	-3.35	1.82	-2.45	1.71
CISC									-2.58	2.41
rack	1.96	0.45	-0.75	0.40	1.02	0.39	1.31	0.39	-0.02	0.42
# of racks	0.04	0.08	-0.12	0.09	0.05	0.08	0.08	0.04	0.05	0.05
ccapac	0.40	0.84	1.04	0.73	0.06	0.85	-0.38	0.20	-0.22	0.30
ccount	-0.41	0.36	2.25	0.88	0.53	0.58	0.12	0.40	0.51	0.56
region dummies										
US	-1.82	0.45	-1.92	0.35	-0.96	0.37	-1.36	0.28	-2.49	0.38
Europe	-0.12	0.27	-0.97	0.18	-0.45	0.16	-0.98	0.28	-1.98	0.38
time dummies										
q196			-1.59	0.69	-1.66	0.53	-1.38	0.66	-0.29	0.68
q296			-0.71	0.59	-1.09	0.52	-0.86	0.67	0.00	0.70
q396			-0.36	0.60	-1.25	0.71	-1.66	0.65	-0.75	0.70
q496			-1.56	0.54	-1.71	0.50	-1.65	0.63	-0.94	0.67
q197			-1.53	0.54	-1.72	0.46	-1.11	0.63	-0.40	0.67
q297			-2.19	0.63	-2.00	0.45	-1.16	0.63	-0.50	0.66
q397			-2.02	0.61	-2.03	0.44	-1.23	0.63	-0.37	0.67
q497			-2.17	0.55	-2.19	0.44	-1.62	0.60	-0.76	0.65
q198	2.07	1.05	-2.33	0.58	-2.63	0.43	-1.75	0.61	-0.93	0.65
q298	1.86	1.19	-2.23	0.53	-2.52	0.42	-2.05	0.59	-0.90	0.64
q398	1.25	1.07	-2.52	0.59	-2.64	0.42	-2.05	0.60	-1.06	0.64
q498	0.65	0.80	-2.83	0.69	-2.84	0.46	-1.84	0.60	-0.89	0.62
q199	0.09	0.64	-3.40	0.75	-3.41	0.50	-2.05	0.59	-1.30	0.60
q299	-0.18	0.57	-3.48	0.83	-3.20	0.54	-2.00	0.56	-0.90	0.65
q399	-0.54	0.52	-3.67	0.84	-3.22	0.58	-2.22	0.55	-1.37	0.65
q499	-1.42	0.47	-4.75	0.96	-3.86	0.72	-2.61	0.57	-1.52	0.67
q100	-1.06	0.47	-4.10	0.96	-3.59	0.71	-2.23	0.59	-1.39	0.68
q200	-1.02	0.49	-4.07	0.89	-3.46	0.62	-2.44	0.57	-1.35	0.65
q300	-0.70	0.50	-3.53	0.81	-3.20	0.59	-2.34	0.56	-1.05	0.67
q400	-1.36	0.50	-3.76	0.84	-3.55	0.72	-2.61	0.55	-1.21	0.66
q101	-1.42	0.50	-3.94	0.89	-3.57	0.71	-2.55	0.55	-1.09	0.67

Table 2: Conditional Logit Estimates for Upgrades

parameter	Linux	s.e.	NetWare	s.e.	NT	s.e.	Unix	s.e.	Other	s.e.
price	-954.15	648.72	-1213.70	342.52	-1225.64	889.09	-96.61	258.44	-76.15	45.50
1st stage R^2		0.89		0.84		0.79		0.81		0.89
firm dummies										
Amdal							0.09	1.02		
AST			-0.06	0.49	0.60	0.87	-0.19	0.58	-0.07	0.90
Compaq	2.47	0.43	2.55	0.26	2.36	0.34	2.12	0.29	1.69	0.33
Dell	1.55	0.30	0.71	0.29	1.27	0.43	0.95	0.31	1.53	0.45
Digital			2.21	0.62	1.68	0.83	1.01	0.76	-0.46	1.02
Fuji-Siemens	1.49	0.71	2.35	0.49	2.20	0.76	1.08	0.43	-0.24	0.83
Gateway	0.16	0.44	0.27	0.30	0.03	0.31	-0.10	0.34	-0.11	0.45
Hewlett-Packard	1.77	0.50	2.11	0.28	1.75	0.29	1.56	0.30	0.95	0.41
Intergraph			-1.51	0.87	0.11	0.47	-0.19	0.88	-0.02	0.62
Micron	0.09	0.82	-0.24	0.36	-0.11	0.34	0.02	0.47	-0.28	0.60
bNCRNetWare			-1.32	0.70	-1.13	0.55	0.05	0.57	2.30	1.78
bsiliNT					-4.48	3.70				
Siemens	2.69	1.04	2.01	0.55	0.18	0.58	0.94	0.57	0.40	0.80
Stratus									4.45	1.49
Sun							3.96	0.58		
Toshiba	0.72	0.68	1.13	0.46	0.88	0.51			1.00	1.81
Unisys	-0.29	1.57	0.30	0.40	1.15	1.10	0.18	0.45	-0.78	0.66
Va Linux	2.70	0.82								
Hitachi	2.54	1.52	2.65	0.68	3.19	1.56	0.12	1.63		
IBM	1.82	0.54	2.19	0.32	2.13	0.52	1.20	0.31	0.81	0.43
Other	1.59	0.50	1.05	0.38	0.79	0.70	1.74	0.53	1.44	0.53
Fuji	2.07	0.88	2.01	0.39	2.56	0.58	1.09	0.64	-0.62	0.55
NEC	1.22	0.55	1.29	0.40	1.14	0.53	-0.30	0.33	-0.38	0.47
Olivetti			0.66	0.54	0.20	0.92	-0.96	0.55	-0.99	0.69
Mitsubishi			-0.43	0.53	0.20	0.55	-1.53	0.82	-0.34	1.16

Note: Each second column denotes a conditional logit regression where the dependent variable is $\pi^u_{j|g} \equiv q^u_j / \sum_{j' \in g} q^u_{j'}$, where q_j is the quantity of upgrades sold of product j, and g denotes the column heading. Number of observations: 4011; 2SLS estimation; instruments listed in Appendix A.3. CISC: dummy for complex instruction set computers; rack: dummy for rack optimized servers; ccapac: potential number of CPUs; ccount: actual number of CPUs.

Table 2 (cont.): Conditional Logit Estimates for Upgrades

parameter	Linux	s.e.	NetWare	s.e.	NT	s.e.	Unix	s.e.	Other	s.e.
characteristics										
PC server	0.84	1.16			1.78	2.42			-3.47	1.26
CISC							-1.81	1.05	3.32	1.20
rack	0.47	0.27	-0.17	0.27	0.03	0.27	-0.19	0.28	0.01	0.31
# of racks	0.07	0.06	0.11	0.03	0.07	0.05	0.03	0.03	-0.05	0.04
ccapac	0.53	0.60	1.09	0.37	1.05	0.92	0.09	0.28	-1.46	0.65
ccount	-0.03	0.25	-0.23	0.21	0.01	0.20	-0.28	0.23	-0.03	0.31
region dummies										
US	-1.75	0.84	-2.31	0.32	-1.43	0.79	-2.44	0.27	-1.42	0.47
Europe	-0.69	0.22	-1.65	0.19	-0.56	0.17	-1.96	0.25	-2.08	0.47
time dummies										
q196			-0.52	0.78	-1.69	0.93	-0.52	0.86	-1.52	0.91
q296			-0.41	0.78	-1.40	1.12	-0.91	0.76	-1.57	0.85
q396			-0.82	0.69	-1.78	1.03	-1.13	0.65	-1.53	0.79
q496			-1.49	0.59	-2.32	0.75	-0.90	0.69	-1.47	0.80
q197			-1.55	0.56	-2.41	0.94	-1.52	0.65	-1.79	0.73
q297			-1.70	0.52	-2.59	0.73	-1.68	0.62	-1.71	0.72
q397			-2.48	0.49	-2.99	0.75	-1.69	0.59	-1.51	0.74
q497			-2.05	0.53	-2.76	0.84	-1.98	0.58	-1.99	0.72
q198	-1.83	0.98	-1.41	0.60	-3.81	0.35	-1.96	0.64	-1.85	0.72
q298	-2.43	0.79	-2.81	0.46	-4.29	0.33	-2.48	0.50	-0.91	0.92
q398	-2.63	0.74	-3.13	0.43	-4.24	0.35	-2.55	0.50	-1.16	0.81
q498	-3.29	0.57	-3.70	0.37	-4.53	0.33	-2.96	0.46	-1.44	0.83
q199	-3.98	0.58	-3.79	0.42	-4.99	0.37	-3.02	0.46	-1.16	0.88
q299	-3.65	0.48	-3.73	0.38	-4.82	0.33	-3.15	0.43	-1.27	0.83
q399	-3.84	0.48	-3.82	0.37	-4.83	0.31	-3.05	0.42	-1.31	0.83
q499	-3.87	0.46	-3.83	0.39	-4.88	0.39	-3.18	0.43	-1.33	0.88
q100	-3.83	0.58	-3.95	0.43	-5.13	0.48	-2.72	0.45	-0.53	0.99
q200	-4.22	0.51	-3.93	0.41	-5.48	0.52	-4.37	0.44	-0.70	0.97
q300	-4.41	0.48	-3.84	0.40	-5.47	0.43	-4.89	0.43	-0.80	0.92
q400	-4.49	0.52	-4.07	0.43	-5.37	0.44	-4.57	0.43	-0.63	0.96
q101	-4.45	0.54	-4.03	0.44	-5.38	0.48	-3.98	0.42	-0.39	1.03

Table 3: Vector Autoregression of Aggregate State Variables

	$\overline{r}^i$	$\overline{r}^u$
$\overline{r}^i(-1)$	0.466	0.106
	(0.28)	(0.16)
$\overline{r}^u(-1)$	0.298	0.686
	(0.26)	(0.15)
constant	0.149	1.418
	(0.86)	(0.49)
adjusted R^2	0.65	0.88
Log likelihood	-2.14	

Note: $\overline{r}^i_t \equiv \sum_{g=1}^5 r_{g,t}$, $\overline{r}^u_t \equiv \sum_{g=1}^5 r^u_{g,t}$; standard errors in parentheses; sample: 1996:1 to 2001:1, missing values for $r_{Linux,t}$, for the period 1996:1 to 2001:1were extrapolated.

Table 4: Polynomial Series Approximations of Inclusive Values

	r_{Linux}	s.e.	$r_{NetWare}$	s.e.	r_{NT}	s.e.	r_{Unix}	s.e.	r_{Other}	s.e.
constant	-10.67	17.91	-9.78	4.45	-0.80	5.11	13.09	10.31	2.13	9.09
$\overline{r}^i$	2.62	18.54	9.28	4.61	0.86	5.29	-13.45	10.67	-2.82	9.41
$\left(\overline{r}^i\right)^2$	0.85	6.25	-2.59	1.55	0.09	1.78	4.64	3.60	0.98	3.17
$\left(\overline{r}^i\right)^3$	-0.21	0.69	0.28	0.17	-0.02	0.20	-0.50	0.40	-0.10	0.35
R^2	0.96		0.99		0.97		0.75		0.66	
	r^u_{Linux}	s.e.	$r^u_{NetWare}$	s.e.	r^u_{NT}	s.e.	r^u_{Unix}	s.e.	r^u_{Other}	s.e.
constant	39.98	17.49	-57.07	15.79	11.34	12.22	-103.54	35.00	64.37	54.12
$\overline{r}^u$	-27.54	11.01	30.91	9.94	-6.13	7.69	68.00	22.03	-39.58	34.07
$\left(\overline{r}^u\right)^2$	6.28	2.29	-5.24	2.07	1.43	1.60	-14.63	4.58	8.22	7.08
$\left(\overline{r}^u\right)^3$	-0.45	0.16	0.30	0.14	-0.10	0.11	1.05	0.31	-0.57	0.49
R^2	0.99		0.99		0.99		0.96		0.59	

Note: $\overline{r}^i_t \equiv \sum_{g=1}^5 r_{g,t}$, $\overline{r}^u_t \equiv \sum_{g=1}^5 r^u_{g,t}$; sample: 1996:1 to 2001:1

Table 5: Maximum Likelihood Estimates

	Static		Optimal Stopping		Full	
parameter	estimate	se	estimate	se		
c	6.706	0.37	6.948	0.14	6.250	0.31
$1-\sigma_{Linux}$	0.636	0.63	0.213	0.29	0.652	0.38
$1-\sigma_{NetWare}$	0.727	0.33	0.196	0.19	0.086	0.22
$1-\sigma_{NT}$	1.213	0.70	0.157	0.26	0.934	0.49
$1-\sigma_{Unix}$	0.204	0.29	0.222	0.52	0.182	0.37
$1-\sigma_{Other}$	0.200	0.42	0.233	1.07	0.028	0.53
c_{Linux}					6.307	0.33
$c_{NetWare}$					6.305	0.32
c_{NT}					6.267	0.33
c_{Unix}					6.244	0.32
c_{Other}					6.205	0.33
$1-\sigma^u_{Linux}$					0.492	0.20
$1-\sigma^u_{NetWare}$					0.330	0.16
$1-\sigma^u_{NT}$					0.326	0.08
$1-\sigma^u_{Unix}$					0.742	0.22
$1-\sigma^u_{Other}$					0.281	0.29
Log likelihood	-0.139		-0.166		-0.105	

Note: Calibrated parameters: $\beta = 0$ for the static model, $\beta = 0.97$ for the optimal stopping and full models; $q = 0.0175$; $ib_0 = 600000$; standard errors calculated by bootstrap, with 1000 replications; sample: 1998:1 to 2001:1

Table 6: Sensitivity to Breakdown Probability

	q=0	q=0.01	q=0.0175	q=0.02	q=0.03	q=0.05	q=0.1	q=0.2	q=0.3	q=0.5
c	6.42	6.32	6.25	6.22	6.13	5.97	5.64	5.19	4.90	4.63
$1-\sigma_{Linux}$	0.67	0.66	0.65	0.65	0.64	0.63	0.61	0.60	0.59	0.58
$1-\sigma_{NetWare}$	0.12	0.10	0.09	0.08	0.07	0.05	0.02	0.00	-0.01	-0.03
$1-\sigma_{NT}$	0.99	0.95	0.93	0.93	0.91	0.87	0.83	0.80	0.78	0.76
$1-\sigma_{Unix}$	0.28	0.22	0.18	0.17	0.13	0.06	-0.04	-0.13	-0.17	-0.20
$1-\sigma_{Other}$	0.15	0.07	0.03	0.01	-0.04	-0.11	-0.22	-0.30	-0.35	-0.39
c_{Linux}	6.45	6.37	6.31	6.29	6.22	6.10	5.85	5.52	5.34	5.38
$c_{NetWare}$	6.45	6.37	6.31	6.30	6.22	6.10	5.85	5.51	5.32	5.35
c_{NT}	6.43	6.34	6.27	6.25	6.17	6.03	5.73	5.29	4.98	4.58
c_{Unix}	6.39	6.30	6.24	6.22	6.14	6.00	5.72	5.32	5.06	4.83
c_{Other}	6.37	6.27	6.20	6.17	6.08	5.91	5.53	4.90	4.33	3.11
$1-\sigma^u_{Linux}$	0.50	0.50	0.49	0.49	0.49	0.48	0.47	0.47	0.48	0.58
$1-\sigma^u_{NetWare}$	0.32	0.33	0.33	0.33	0.34	0.35	0.39	0.45	0.51	0.65
$1-\sigma^u_{NT}$	0.34	0.33	0.33	0.33	0.32	0.31	0.30	0.28	0.27	0.26
$1-\sigma^u_{Unix}$	0.73	0.74	0.74	0.74	0.75	0.76	0.79	0.84	0.88	0.96
$1-\sigma^u_{Other}$	0.26	0.27	0.28	0.29	0.31	0.35	0.49	0.69	0.64	-0.21
Log likelihood	-0.103	-0.104	-0.105	-0.105	-0.106	-0.107	-0.108	-0.110	-0.111	-0.111

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

Table 7: Sensitivity to Discount Factor

	β=0.9	β=0.95	β=0.97	β=0.98	β=0.99
c	6.08	6.20	6.25	6.27	6.29
$1-\sigma_{Linux}$	0.65	0.65	0.65	0.65	0.65
$1-\sigma_{NetWare}$	0.08	0.08	0.09	0.09	0.09
$1-\sigma_{NT}$	0.93	0.93	0.93	0.93	0.94
$1-\sigma_{Unix}$	0.18	0.18	0.18	0.18	0.18
$1-\sigma_{Other}$	0.01	0.02	0.03	0.03	0.03
c_{Linux}	6.35	6.32	6.31	6.31	6.30
$c_{NetWare}$	6.33	6.32	6.31	6.31	6.31
c_{NT}	6.22	6.26	6.27	6.28	6.29
c_{Unix}	6.21	6.23	6.24	6.25	6.25
c_{Other}	6.07	6.16	6.20	6.22	6.23
$1-\sigma^u_{Linux}$	0.50	0.49	0.49	0.49	0.49
$1-\sigma^u_{NetWare}$	0.33	0.33	0.33	0.33	0.33
$1-\sigma^u_{NT}$	0.31	0.32	0.33	0.33	0.33
$1-\sigma^u_{Unix}$	0.73	0.74	0.74	0.74	0.74
$1-\sigma^u_{Other}$	0.10	0.23	0.28	0.31	0.33
Log likelihood	-0.105	-0.105	-0.105	-0.105	-0.105

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

Table 8: Sensitivity to Initial Installed Base

	ib_0=0	ib_0=600	ib_0=6000	ib_0=600000	ib_0=1000000	ib_0=1500000
c	5.67	5.67	5.68	6.25	6.55	6.89
$1-\sigma_{Linux}$	0.65	0.65	0.65	0.65	0.66	0.66
$1-\sigma_{NetWare}$	0.08	0.08	0.08	0.09	0.09	0.10
$1-\sigma_{NT}$	0.92	0.92	0.92	0.93	0.94	0.96
$1-\sigma_{Unix}$	0.15	0.15	0.15	0.18	0.20	0.23
$1-\sigma_{Other}$	0.01	0.01	0.01	0.03	0.05	0.08
c_{Linux}	5.70	5.70	5.70	6.31	6.64	7.01
$c_{NetWare}$	5.70	5.70	5.71	6.31	6.64	7.02
c_{NT}	5.66	5.66	5.67	6.27	6.60	6.97
c_{Unix}	5.62	5.62	5.63	6.24	6.57	6.94
c_{Other}	5.58	5.58	5.59	6.20	6.53	6.90
$1-\sigma^u_{Linux}$	0.46	0.46	0.46	0.49	0.52	0.57
$1-\sigma^u_{NetWare}$	0.32	0.32	0.32	0.33	0.35	0.38
$1-\sigma^u_{NT}$	0.24	0.24	0.25	0.33	0.37	0.42
$1-\sigma^u_{Unix}$	0.69	0.69	0.69	0.74	0.78	0.82
$1-\sigma^u_{Other}$	0.99	0.99	0.97	0.28	0.23	0.29
Log likelihood	-0.108	-0.108	-0.108	-0.105	-0.103	-0.099

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

C Figures

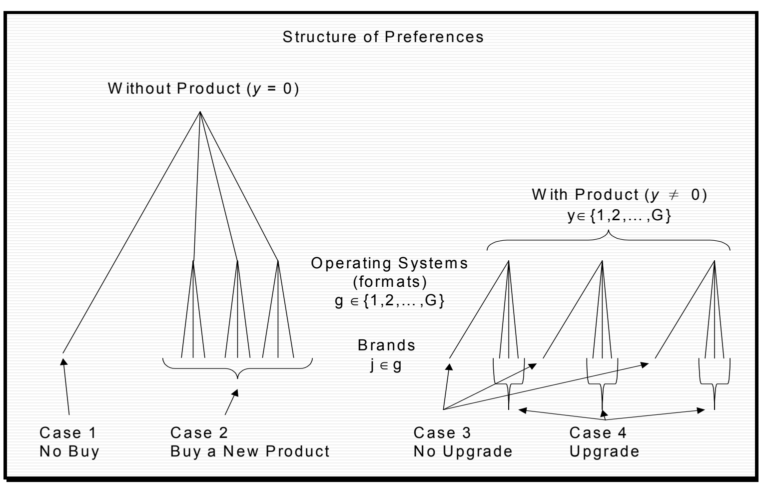

Figure 1: Preference Structure

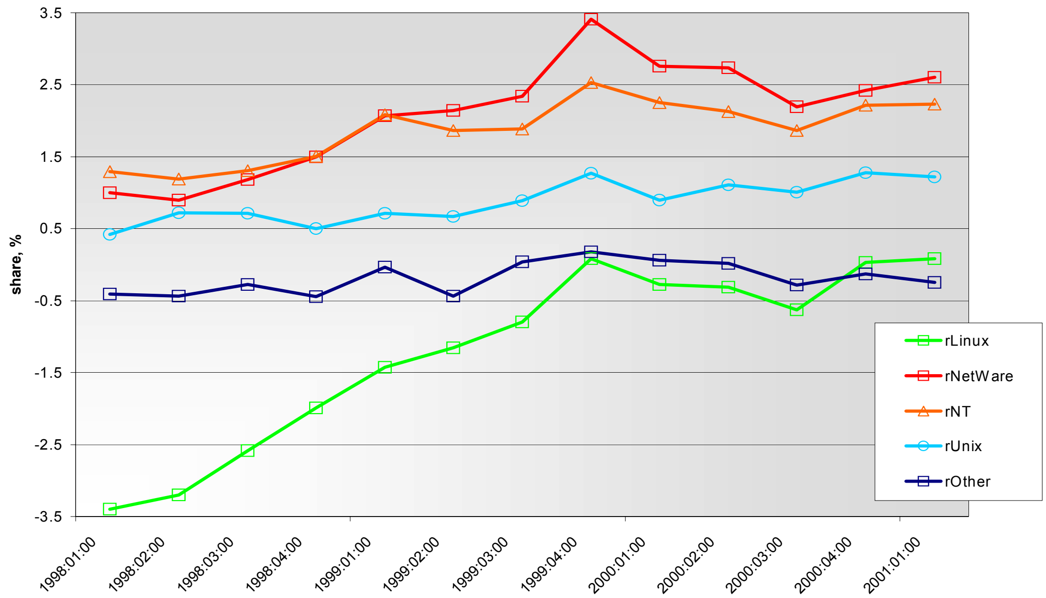

FIGURE 2: ISS inclusive values $r_{g,t}$, g =Linux, Netware, Windows NT, Unix, Other.

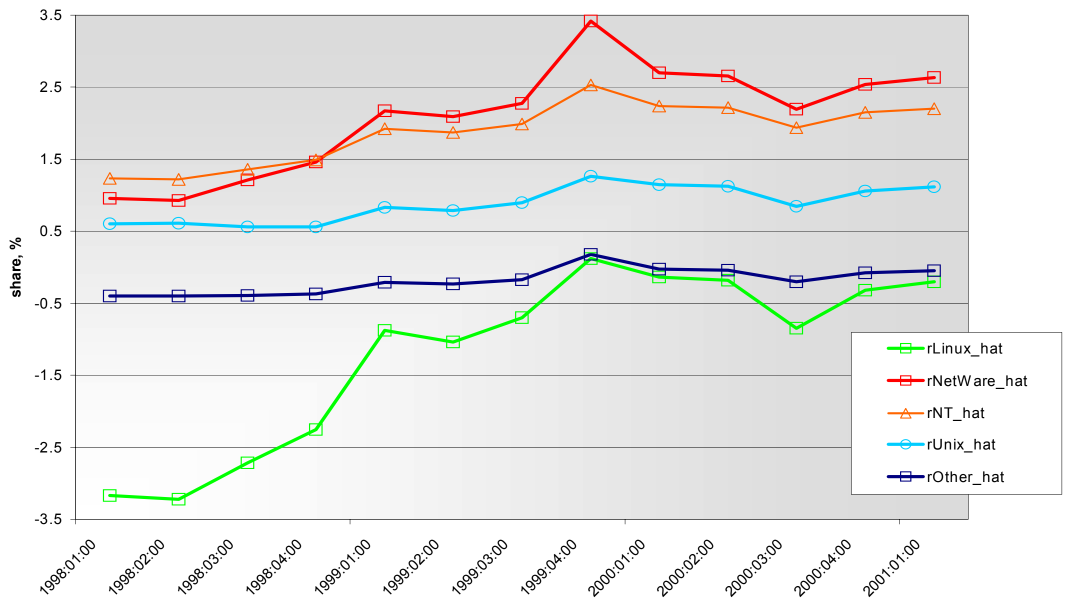

FIGURE 3: Polynomial approximations of ISS inclusive values: $\widehat{r}_{g,t}$, g =Linux, Netware, Windows NT, Unix, Other.

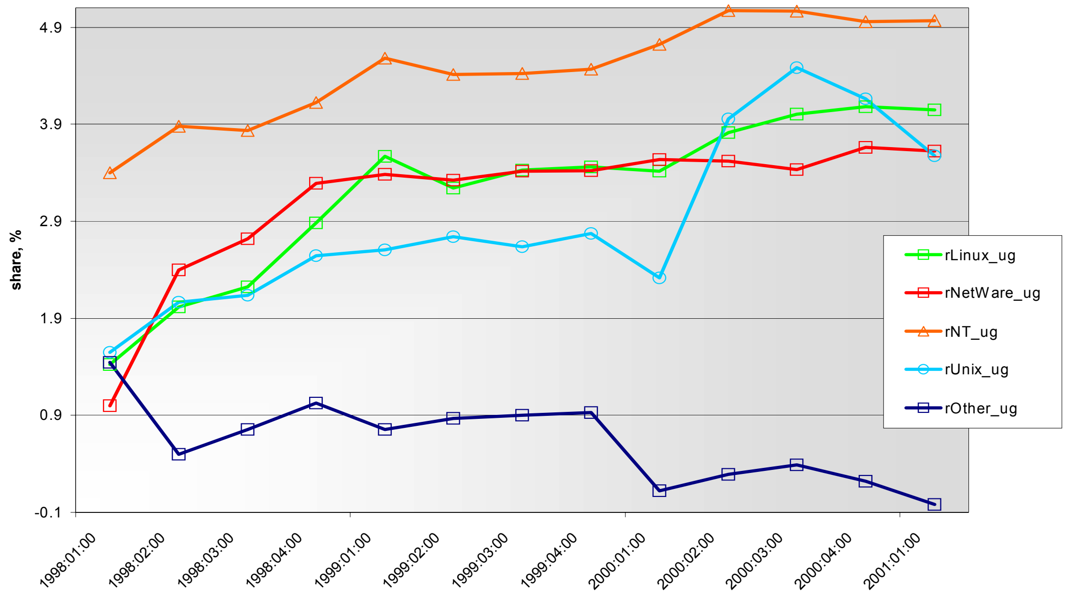

FIGURE 4: Upgrade inclusive values $r^u_{g,t}$, g =Linux, Netware, Windows NT, Unix, Other.

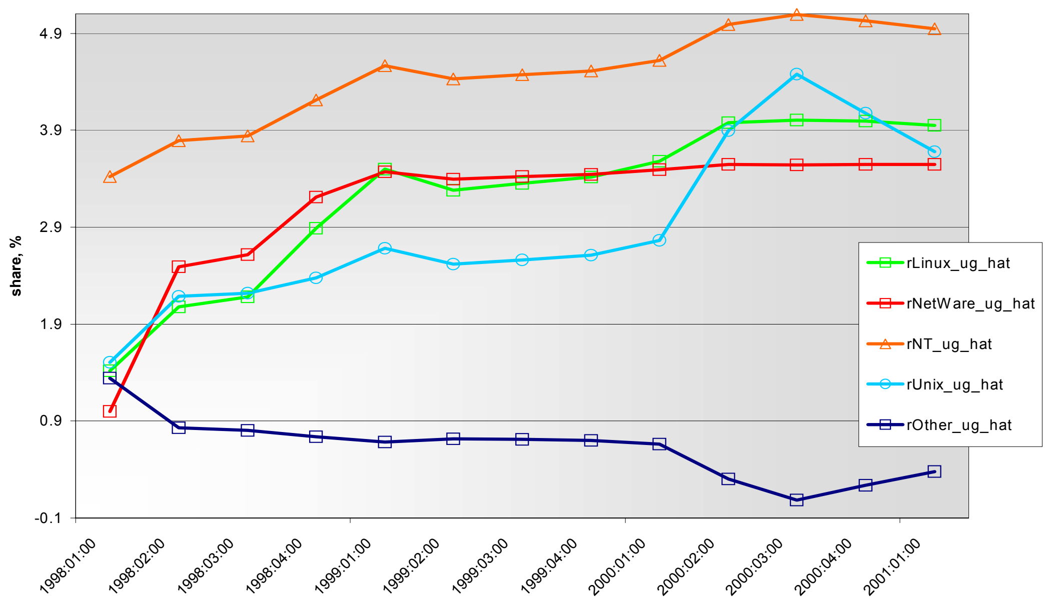

FIGURE 5: Polynomial approximations of Upgrade inclusive values: $\widehat{r}^u_{g,t}$, g =Linux, Netware, Windows NT, Unix, Other.

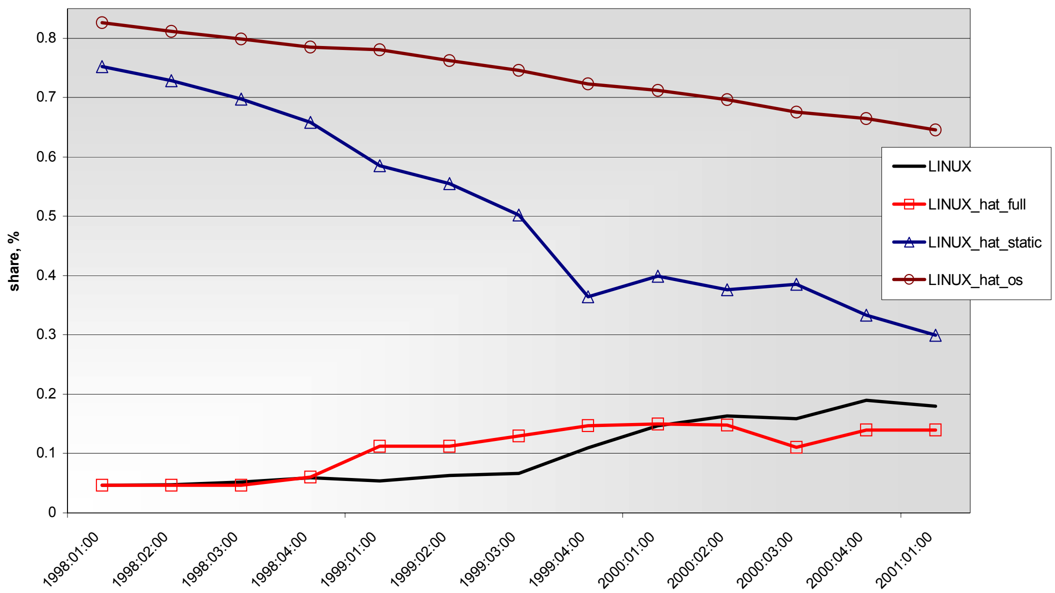

FIGURE 6: Linux actual and predicted shares $\pi_{Linux,t}$ and $\widehat{\pi}_{Linux,t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.

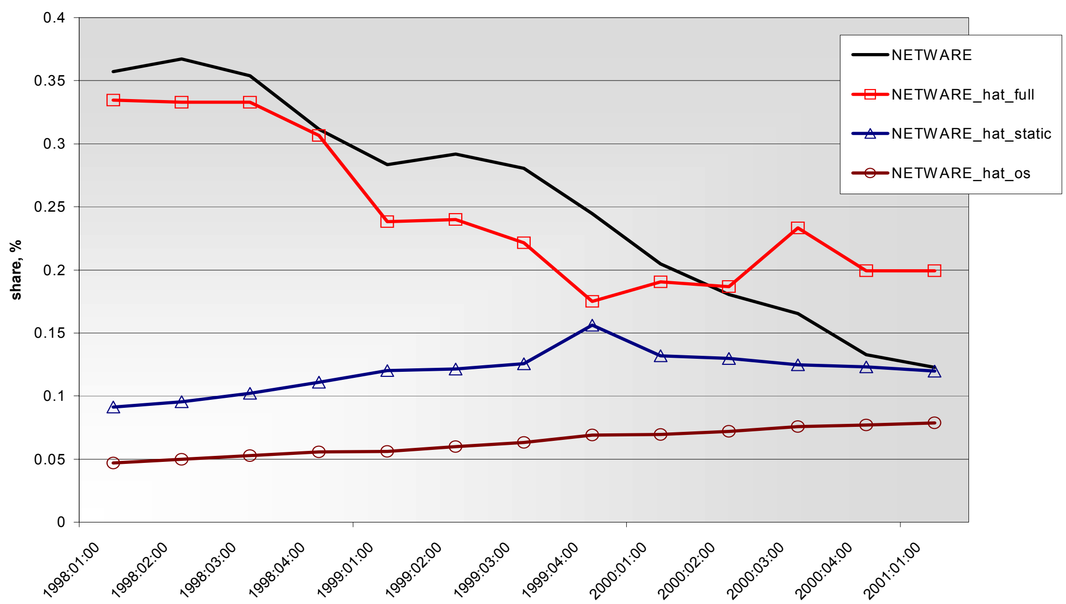

FIGURE 7: NetWare actual and predicted shares $\pi_{NetWare,t}$ and $\widehat{\pi}_{NetWare,t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.

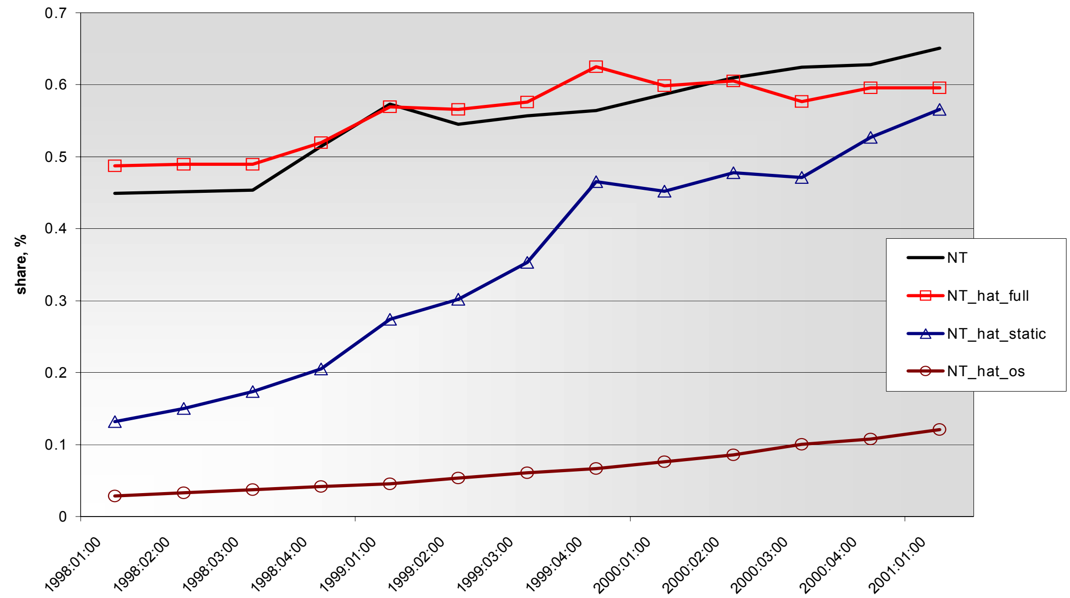

FIGURE 8: Windows NT actual and predicted shares $\pi_{NT,t}$ and $\widehat{\pi}_{NT,t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.

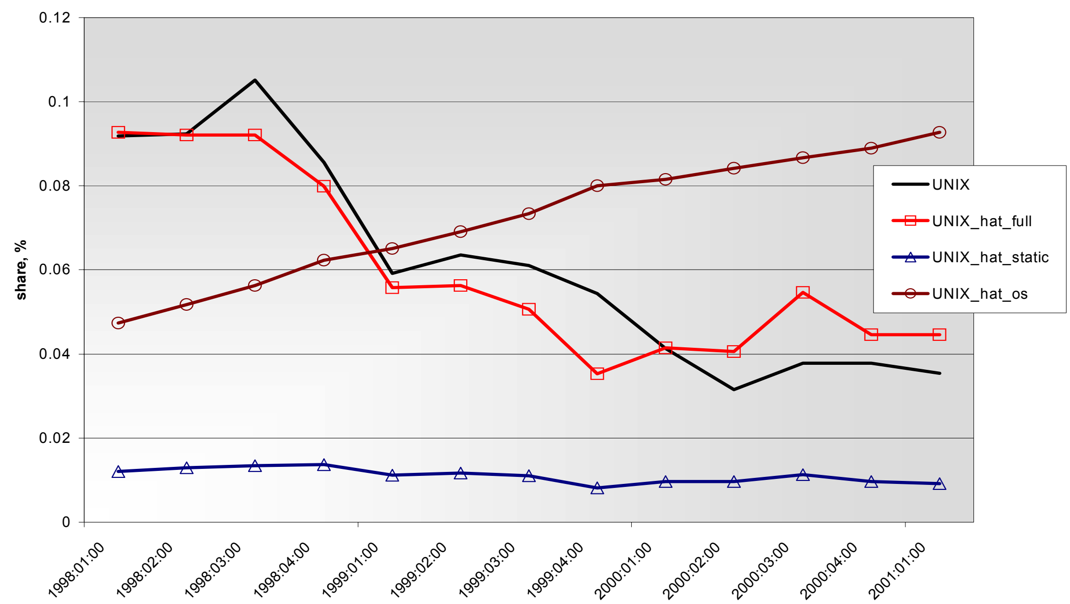

FIGURE 9: Unix actual and predicted shares $\pi_{Unix,t}$ and $\widehat{\pi}_{Unix,t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.

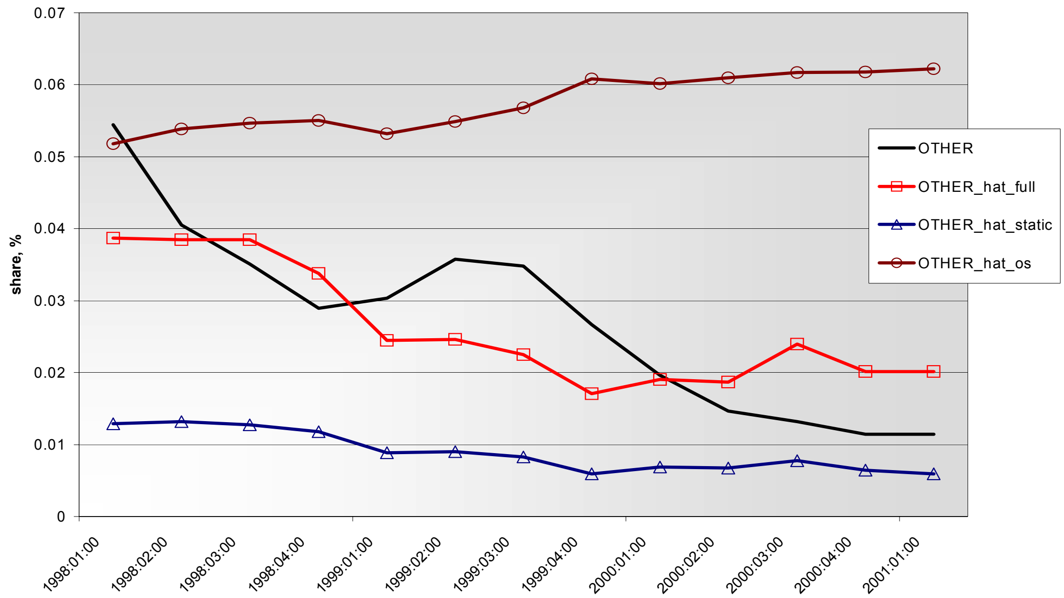

FIGURE 10: Other OSs actual and predicted shares $\pi_{Other,t}$ and $\widehat{\pi}_{Other,t}$. Results from three models: the full dynamic model, a simple optimal stopping model and a static one.

References

Ackerberg, D., (2003): "Advertising, Learning, and Consumer Choice in Experience Good Markets: A Structural Empirical Examination," *International Economic Review*, Vol. 44, No. 3, 1007-1040.

Aguirregabiria, V., (2002): "Sales promotions in supermarkets: Estimating their effects on profits and consumer welfare," *mimeo*, Boston University.

Aguirregabiria, V., P. Mira, (2002): "Swapping the Nested Fixed Point Algorithm: A Class of Estimators for Discrete Markov Decision Models," *Econometrica*, Vol 70, 4, 1519-1543.

Aguirregabiria, V., P. Mira, (2004): "Sequential Estimation of Dynamic Discrete Games," *mimeo*, Boston University and CEMFI, Madrid.

Bajari, P., C. L. Benkard, J. Levin, (2005): "Estimating Dynamic Models of Imperfect Competition," *mimeo.*

Berry, S. T., (1994): "Estimating Discrete-Choice Models of Product Differentiation," *Rand Journal of Economics*, Vol 25, 2, Summer, 242-262.

Berry, S. T., J. Levinsohn, A. Pakes, (1995): "Automobile Prices in Equilibrium" *Econometrica*, Vol 63, 4, 841-890.

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.

Berry, S. T., A. Pakes, (2000): "Estimation from the Optimality Conditions for Dynamic Controls," *mimeo*, Yale University and Harvard University.

Cardell, N. S., (1997): "Variance Components Structures for the Extreme-Value and Logistic Distributions with Application to Models of Heterogeneity," *Economic Theory*, 13, 185-213.

Carranza, J. E., (2005): "Product Innovation in Durable Goods Markets," *mimeo*, University of Wisconsin-Madison.

Cho, S.-J., (2002): "An Empirical Model of Mainframe Computer Investment," *mimeo*, Yale University.

Ericson, R, A. Pakes, (1995): "Markov Perfect Industry Dynamics: A Framework for Empirical Work," *Review of Economic Studies*, vol. 62, no. 1, pp. 53-82.

EU Commission, (2004): "Commission Decision of 24.03.2004 relating to case COMP/C-3/37.792 Microsoft."

Gordon, B. R., (2005): "Estimating a Dynamic Model of Demand for Durable Goods," *mimeo*, Carnegie Mellon University.

Hendel, I., A. Nevo, (2003): "Measuring the Implications of Sales and Consumer Stockpiling Behavior," *mimeo*, University of Wisconsin and University of California.

Hotz, J., R. A. Miller, (1993): "Conditional Choice Probabilities and Estimation of Dynamic Models," *Review of Economic Studies*, 60, 497-529.

IDC, (1998): "Bulletin. IDC's Server Market Methodology, Database Structure, and Definitions."

Ivaldi, M., Sz. Lőrincz, (2005): "A Full Equilibrium Relevant Market Test: Application to Computer Servers," *IDEI working paper.*

Ivaldi, M., F. Verboven, (2004): "Quantifying the Effects from Mergers in European Competition Policy," *International Journal of Industrial Organization*, forthcoming.

McFadden, D., (1981): "Econometric Models of Probabilistic Choice," in C.F. Manski, D. McFadden (eds), *Structural Analysis of Discrete Data with Econometric Applications*, MIT Press, Cambridge, Massachusetts, 198-272.

Melnikov, O., (2000): "Demand for Differentiated Durable Products: The Case of the U.S. Computer Printer Market," *mimeo*, Yale University.

Pakes, A., (2000): "A Framework for Applied Dynamic Analysis in I.O.," *NBER working paper* #8024.

Pakes, A., P. McGuire, (1994): "Computing Markov Perfect Nash Equilibrium: Numerical Implications of a Dynamic Differentiated Product Model," *RAND Journal of Economics*, vol. 25, no. 4, pp. 555-589.

Park, S., (2004): "Quantitative Analysis of Network Externalities in Competing Technologies: The VCR Case ," *Review of Economic Studies*, Vol. 86, 4, 937-945.

Pesendorfer, M., P. Schmidt-Dengler, (2004): "Least Squares Estimators for Dynamic Games," *mimeo*, London School of Economics.

Rust, J., (1987): "Optimal Replacement of GMC buses: An Empirical Model of Harold Zurcher," *Econometrica*, Vol. 55, 5, 999-1033.

RUST, J., (1994): "Structural Estimation of Discrete Decision Processes," in R.F. Engle, D.L. McFadden (eds), *Handbook of Econometrics, Volume IV*, Elsevier Science, 3081-3143.

RUST, J., (1997): "Using Randomization to Break the Curse of Dimensionality," *Econometrica*, Vol. 65, 3, 487-516.

VAN REENEN, J., (2003): "Demand and Prices in the Work Group Server Market," *mimeo*, London School of Economics.