Designing a Better Shopbot

by

Alan L. Montgomery, Kartik Hosanagar, Ramayya Krishnan, and Karen B. Clay

June 2001

Alan L. Montgomery (e-mail: alan.montgomery@cmu.edu) is an Associate Professor at the Graduate School of Industrial Administration, Kartik Hosanagar (email: kartik.hosanagar@andrew.cmu.edu) is a doctoral student, Ramayya Krishnan (email: rk2x@andrew.cmu.edu) is the William W. and Ruth F. Cooper Professor of Management Science and Information Systems, and Karen B. Clay (email: kclay@andrew.cmu.edu) is an Assistant Professor of Economics at the Heinz School, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213.

Abstract:

A primary tool that consumers have for comparative shopping are shopbots, which is short for shopping robots. These shopbots automatically search a large number of vendors for price and availability. However, the vast majority of Internet shoppers do not use them. One explanation for this low usage rate is the poor design of many shopbots. Typically the shopbot searches a predefined set of vendors and reports all results. This can result in time consuming searches that provide redundant or dominated alternatives. Our research demonstrates analytically how shopbot designs can be improved by developing a utility model for consumer purchases. This utility model considers the intrinsic value of the product and its attributes, the disutility from waiting, and the cognitive costs associated with evaluating the offers retrieved. We focus on the operational decisions made by the shopbot: which stores to search, how long to wait, and which offers to present to the user. To illustrate our model we calibrate the model to price and response time data collected at online bookstores over a six-month period. Using prior expectations about price and response time can substantially increase the utility consumers receive from using shopbots. Specifically, we demonstrate how shopbots can search more intelligently and increase consumer utility by selectively presenting results.

Keywords: Computer agents, utility theory, informational retrieval, stochastic modeling

1 Introduction

A primary tool that consumers have for comparison shopping on the Internet is comparison-shopping engines, also known as shopping robots or shopbots. These shopbots automatically search a large number of vendors for price and availability. Given the wide degree of price variation for consumer products on the Internet, comparison shopping can provide real benefits (Hann, Hitt, and Clemens 2000; Brynjolfsson and Smith 2000; Clay, Krishnan, and Wolff 2001). For example, suppose a consumer wishes to purchase the novel *Bear and the Dragon* by Tom Clancy. A visit to *dealtime.com* during October 2000 resulted in 57 offers that range in price from \$16.45 to \$40.22 after a delay of about 30 seconds.

Despite their apparent usefulness, most consumers continue to search in the traditional way, e.g., visiting a single store or selected sample of stores and making comparisons on their own. Media Metrix reports that during July 2000 less than 4% of Internet users used a shopbot, while over 67% visited an online retailer (28% visited an online bookstore). An obvious explanation is that consumers are simply not aware of shopbots, which could be rectified through increased advertising. Another explanation which forms the focus of our research is that consumers may prefer the traditional shopping process over shopbots. In other words simply visiting your favorite store like Amazon is better than using a shopbot even though the shopbot may include a search to Amazon. To an economist this would imply that the utility from a traditional shopping process exceeds the utility generated by a shopbot.

To explore this possibility we develop an analytical model of consumer utility to weigh the expected gains of search (e.g., lower prices or higher utility) against the costs (e.g., waiting time for the shopbot to respond and cognitive effort required to compare alternatives). Using this model, the shopbot can make the following operational decisions that influence these benefits and costs: which stores to query for offers, how long to wait for these stores to respond, and which items to report to a user. This benefit-cost framework allows us to evaluate the utility of the traditional shopping process, current shopbot design, and optimal shopbot design. We show that the traditional shopping process may be preferred by consumers over current shopbots because they are too slow or present too many alternatives relative to the expected gains of search. More importantly, we show how the shopbot's operational decisions impacts consumer utility and consider the implications for improved shopbot design.

Expectations about prices are an important element in our improved shopbot design. The current generation of shopbots are continually retrieving information about prices, yet never learn about the dispersion of prices from these searches. For example, Amazon may have a high probability of being cheap for bestsellers, but Fatbrain may have a high probability of being cheaper for computer books. We improve a shopbot's decision about where to search by incorporating prior expectations about prices. Using data collected from 28 stores between August 1999 and January 2000, we find that shopbots can predict prices with a high degree of accuracy. These predictions can be used to select which stores to query.

In summary, our model shows that simply searching all stores and presenting consumers an ordered list of all vendors is a poor design. Shopbots can be dramatically improved by incorporating knowledge about how consumers make judgements and the value they attach to price search. This shows that a utility based approach to shopbot design has merit. In §2 we discuss the online shopping process for books and the shopbot's operational decisions. §3 develops an analytical model of optimal shopbot design that incorporates a consumer utility model to balance tradeoffs between attributes like price, delivery, and quality against waiting time. In §4 we develop statistical models to predict price and describe response time to retrieve offers from online bookstores using data collected over a six-month period at online bookstores. §5 presents the results of a simulation analysis that demonstrates the gains from this new type of shopbot compared with current shopbot designs. The impact of these results on profitability for the shopbot is discussed in §6. We conclude the paper in §7 with a discussion of our findings and directions for future research.

2 Operational Decisions in Shopbot Design

The typical shopping process for a consumer begins with a consumer identifying one or more books that she is interested in purchasing. (In this paper we will focus on the purchase of books since it is the most widely purchased item on the Internet, although our technique is applicable to travel and other consumer goods as well.)

Identification of the book may happen through a book review or by browsing an online bookstore, physical bookstore, a shopbot, or a mix of the above approaches. Evidence suggests that many people do not engage in search. In fact the vast majority of online bookstore consumers simply visit a single bookstore (Johnson et al 2000b).

Currently shopbots query all stores at the time a request is made and report all results to consumers[1]. Query time can be substantial, with the modal time for pricescan and dealpilot being about 45 seconds. The tails of the distribution are fat, especially for dealpilot which times out at 3 minutes in 10 percent of the searches. In comparison, individual stores respond more rapidly, with modal response times of 2 seconds, but also have the potential for slow response. Our conjecture is that many consumers may choose to go directly to an online bookstore and avoid a shopbot because they are impatient. Usability research shows that delays of more than 10 seconds result in a loss of user attention (Nielsen 2000, pg. 44). Instead of querying all stores, shopbots could use prior expectations about prices to identify stores that are most likely to have low prices or high utility. More sophisticated shopbots could reduce both the average and the tails of their distribution by reducing the number of queries or interrupting searches, making shopbots more attractive to consumers.

An additional problem with current shopbot design is the number of alternatives that are presented. Every additional alternative presented will force the user to expend cognitive effort. Given that consumers are cognitive misers, additional time spent on cognitive activity is more taxing than simply waiting. One suggestion would be to show consumers the cheapest offer. However, it is unlikely that the shopbot could be so confident since there is a random component to utility. Nor is the best offer synonymous with the cheapest offer, since other attributes such as delivery and the store's identity may effect a consumer's utility. We propose that shopbots could sort the retrieved offers by utility and offer a consumer fewer but more relevant choices by eliminating unlikely alternatives. Fewer alternatives reduce the cognitive burden to the consumer, however this

---

1. Some shopbots may have direct access to price information from stores due to special marketing agreements. In these circumstances query time may be substantially lessened since it requires a lookup in a local database as opposed to querying a store and appropriately parsing the HTML document that is returned.

reduction also comes at an expense. The utility of the offer set necessarily declines with the number of options.

For our purposes we assume that our customer is interested in searching using a shopbot. A shopbot presents the consumer with choices such as location (country), state (for sales tax purposes), and currency. Once the consumer makes her choices and initiates the search, the shopbot queries all of the bookstores in its search set and tabulates the information from stores that respond within a specified period. The shopbot then presents the consumer with prices and shipping options ranked by total price. Shopbots may filter these stores for availability and add information about the online bookstores (e.g., MySimon.com also provides Gomez ratings). Note that if a consumer prefers that the data be ranked by some other criteria, she can re-rank it based on other data fields such as unit price, store name, delivery type, and so forth.

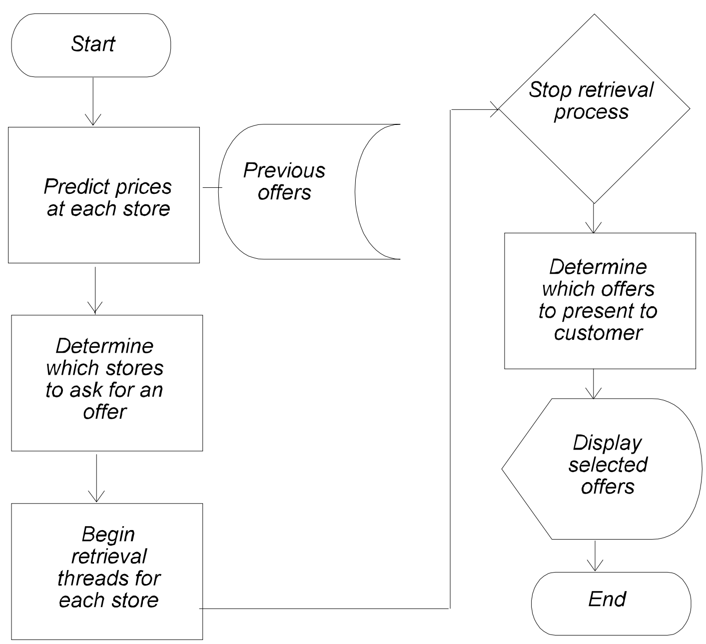

Figure 1. Flow-diagram illustrating the operational decision process for a shopbot.

Figure 1 illustrates the decision framework we assume for the shopbot's operational decision. First, the shopbot is given the book to search. Second, the shopbot makes predictions about the price and response time at each store it will consider querying. These prior predictions can be used to determine which stores to search and how long to wait for a response. We assume that only realized offers (i.e., those offers retrieved from a store) can be presented to consumers. This forces shopbots to actively search and not rely on prior expectations[2]. Finally, once the process threads[3] that search the stores are started, the shopbot must decide whether to wait for all threads to finish their retrieval or whether to prematurely stop the retrieval process. Once all offers are collected, the shopbot decides which of these offers to present to the customer. Separating the presentation and query decisions allows the shopbot to respond to poor initial predictions.

3 A Utility Model of Consumer Interaction with a Shopbot

In this section we formalize the operational decisions made by the shopbot that were introduced in the last section. We begin in §3.1 by presenting a random utility model which allows us to quantify the value of a set of offers and balance this value against the cognitive effort necessary to compare this set. A model of the shopbot decision process is given in §3.2. The optimal shopbot design is considered in §3.3 and §3.4.

We assume that there is a universe of S stores that can be queried by the shopbot. The shopbot must choose which stores to query and how many seconds (t^*) to allow the query to run. The decision of which stores to query ($\boldsymbol{q}$) is encoded as an Sx1 vector of binary decision variables, where q_i=1 if the store is to be queried and q_i=0 otherwise. The time it takes for the ith store to respond is assumed to be a random variate T_i and the observed value is t_i. The corresponding Sx1 vector of observed response times is $\boldsymbol{t}$. The Sx1 vector $\boldsymbol{r}$ records whether the store responds to the query within t^* seconds, that is r_i=1 if $t_i<t^*$ and r_i=0 otherwise. Notice that $\boldsymbol{r}$ is a random variate that is not chosen by the shopbot directly, but is a function of realized retrieval times ($\boldsymbol{t}$) and the shopbot's choice of $\boldsymbol{q}$ and t^*. For example, suppose the ith alternative has been queried, q_i=1, if the realized

2. This assumption could be relaxed so that expected offers could be shown to the consumer. However, this adds an additional layer of complexity since the consumer must now evaluate the probability that an offer will be available.
3. A process thread refers to a task within a program that can be run independently from the main body of the program. Most operating systems allow multiple threads to be run at the same time. In our shopbot example the query for each store would be launched as a separate thread.

time is less than the time to cancel all threads, $t_i < t^*$, then the offer will be retrieved, $r_i=1$, otherwise even though the store was queried its offer will not be retrieved, $r_i=0$.

Once this query is completed the shopbot decides which offers to present to the customer. We define the vector $\boldsymbol{p}$ to represent which offers to present, where $p_i=1$ if the offer is presented to the consumer and $p_i=0$ otherwise. The total number of stores queried, stores retrieved, and offers presented equal Q, R, and P, respectively. The sum of the vector elements yield the number of chosen items, $P=\sum p_i$, $Q=\sum q_i$, and $R=\sum r_i$. Notice the following inequality holds $P \leq R \leq Q \leq S$. We assume that only those stores that are queried can be retrieved, and only those offers that are retrieved can be offered. This relationship can be represented by the inequality $\boldsymbol{p} \leq \boldsymbol{r} \leq \boldsymbol{q}$, which is defined in terms of the element-wise inequalities, $0 \leq p_i \leq r_i \leq q_i \leq 1$. The requirement that only retrieved offers are presented reflects current shopbot design, but could be relaxed in future research to allow predicted offers to be reported.

3.1 A Random Utility Model

The utility from ith store is a random variate, U_i, and $\boldsymbol{U}=\{U_1, U_2, ..., U_S\}$. We argue that the consumer will choose only one item within the set, and therefore the utility from a set of choices is equal to the utility from the best alternative or the maximum of the set. We define the operator $\boldsymbol{U}\langle\boldsymbol{p}\rangle$ to denote the set comprised of the elements of the set $\boldsymbol{U}$ that correspond with those elements in $\boldsymbol{p}$ that equal one. Additionally, we define the operator $\langle\langle\boldsymbol{p}\rangle\rangle$ to denote the set of the indices that correspond to the non-zero elements of $\boldsymbol{p}$. For example, if $\boldsymbol{p}=[1\ 1\ 0\ 0\ 1]$ then $\langle\langle\boldsymbol{p}\rangle\rangle=\{1, 2, 5\}$ and $\boldsymbol{U}\langle\boldsymbol{p}\rangle=\{U_1, U_2, U_5\}$.

The utility of the ith offer is modeled (U_i) as the sum of the utility derived directly from the product ($\mathring{U}_i$) and the disutility associated with the waiting time for the online stores to respond to the shopbot's query (W), the overhead of launching Q threads on the shopbot, and the cognitive effort (C) associated with evaluating the set of alternatives:

$$U_i = \mathring{U}_i - \xi W - \omega Q - \lambda C \ . \quad (1)$$

Since the latter three terms decrease utility, we assume $\xi, \omega, \lambda > 0$. Additionally, these disutility terms are not subscripted by *i* since they are identical for all items offered. They are a function of the set of offers queried and presented. We consider the construction of each of these variables below.

Product utility: We assume that product utility ($\mathring{U}_i$) is the sum of a component due to a linear function of the attributes of the product ($\bar{U}_i$) and a stochastic component (ϵ_i):

$$\mathring{U}_i = \bar{U}_i + \epsilon_i , \quad \text{where } \bar{U}_i = \sum_{j=1}^{A} \beta_{ij} a_{ij} , \tag{2}$$

and β_{ij} denotes the weight and a_{ij} denotes the value of the *j*th attribute of the *i*th product. For example, in our application to bookstores the attributes of a book are price, delivery, tax, shipping time, and store name.

This compensatory utility model allows increases in one attribute to offset the decrease associated with another attribute versus a non-compensatory approach that assumes consumers have rigid thresholds or reservation prices. A primary benefit of the compensatory approach is that it can better capture tradeoffs that consumers make. For example, the increased price of a book at Amazon may be offset by their guarantee of a shorter delivery time.

The stochastic component is due to unobservable factors or random evaluation error by the consumer and represents the fact that we cannot predict utility with certainty. We assume that these ϵ_i are independently and identically distributed and follow an extreme value distribution with a zero location parameter and a scale parameter of θ. The cumulative distribution of $\mathring{U}_i = \bar{U}_i + \epsilon_i$ is:

$$Pr[\mathring{U}_i \leq x] = \exp\{- e^{-(x-\bar{u}_i)/\theta}\}. \tag{3}$$

The mean and variance are $E[\mathring{U}_i] = \bar{U}_i + \gamma\theta$ and $Var[\mathring{U}_i] = \pi^2\theta^2/6$, where γ is Euler's constant (i.e., $\gamma \approx .57722$). The choice of the extreme value distribution is motivated by its extensive use in choice models (McFadden 1980)[4]. Usually the scaling parameter θ is set to unity to address the identification of the parameter estimates in the utility function when a constant is included.

4. The IIA property of the choice process implied by extreme value distribution is a reasonable one in this instance, since all the products in the set are considered to be close, if not perfect, substitutes. The addition of another similar item in this set will reduce the choice probability of the other alternatives in the set.

Waiting Time: The response time (W) is the time associated with retrieving the set of offers. This is the time for the slowest store to respond, max($\boldsymbol{T}$<$\boldsymbol{q}$>), unless this time exceeds the interrupt time set by the shopbot when the retrieval threads are launched, t^*, in which case any remaining threads will be ignored. Hence, response time (W) is defined as

$$W = \min(t^* , \max(\boldsymbol{T}\langle\boldsymbol{q}\rangle)) \ . \tag{4}$$

Server Overhead: We assume that the time for the shopbot's server to start and service the threads that handle the HTTP query to an online store is proportional to the number of stores queried (Q) and is measured by ω. The term ωQ measures the disutility that consumers experience as the result of the total delay in response time from launching Q threads. It captures only the opportunity costs borne by the consumer and does not represent the shopbot's computational cost of the search which is irrelevant to the consumer. For most situations one would expect that ω would be insignificant, since the time to start a thread is on the order of a few milliseconds, hence $\omega \approx 0$. However, when the system is above its operating capacity during peak periods there could be a measurable delay for the consumer (e.g., perhaps a second or more). An alternative interpretation of ω is delay due to the load on the shopbot's server. The inclusion of $\omega > 0$ prevents the situation in which the optimal shopbot design would be for the shopbot to launch a query to every store that is known. Under this solution the shopbot server's would always be overutilized and response seriously degraded.

Cognitive Costs: A metric for evaluating the cognitive costs (C) associated with comparing P alternatives each with A attributes was proposed by Shugan (1980):

$$C = (A - 1)(P - 1) \ . \tag{5}$$

The motivation for this formulation is that cognitive costs are proportional to the number of alternative and attribute pairs. For example, a set of three alternatives with four attributes will take six comparisons.

3.2 The Shopbot's Decision Problem

The shopbot's decision problem is to maximize utility in a two step process. First, the shopbot needs to make decisions about which stores to query ($\boldsymbol{q}$) and how long to wait (t^*) in order to maximize the expected utility of the offers ($\boldsymbol{p}$) that it expects to make to a consumer:

$$\max_{\boldsymbol{q},t^*} \; E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle)] \; . \tag{6}$$

At this stage the shopbot does not know prices nor utility with certainty, instead it must predict utility and its components like price, delivery time, etc. using past information. Additionally, the shopbot does not need to decide which stores it will present since this decision can be made after the offers are retrieved, but it does need to predict which $\boldsymbol{p}$ is likely to be used.

Second, after t^* seconds have elapsed and the shopbot has actually retrieved a set of offers ($\boldsymbol{r}$) it needs to make a decision about which offers to present ($\boldsymbol{p}$) to the consumer:

$$\max_{\boldsymbol{p}} \; E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid \bar{\boldsymbol{U}}\langle\boldsymbol{r}\rangle] \; . \tag{7}$$

Notice that at this point the shopbot knows the portion of utility due to the product attributes ($\bar{U}$) but not the portion due to the evaluation error (ϵ_j). As an aid to the reader the variables and parameters used in this paper are listed in Table 1.

This problem presents a large decision space in which the shopbot can operate out of which current shopbot design considers only one particular combination. In our notation current shopbot design can be represented as a query to all stores ($\boldsymbol{q}=\iota$) where $\iota=[1\ 1\ \ldots\ 1]'$, interrupt the search if it is not completed within 30 seconds ($t^*=30$), and present all retrieved offers ($\boldsymbol{p}=\boldsymbol{r}$). However, there are 2^S possible combinations that the shopbot could consider for query. In addition for any proposed retrieval set there are many combinations from which to show consumers. In total, there are $\sum_{i=1}^{S}\sum_{j=1}^{i}\binom{S}{i}\binom{i}{j}$ possible sets that the shopbot could offer the consumer. A universe of 10 stores yields 58,025 combinations, while a universe of 30 stores yields more than 205 trillion combinations. In the next two subsections we consider the solution to this problem. Since this is a two step optimization problem we begin with the final stage decision and then consider the initial decision assuming the final stage decision will be optimized.

Notation	Description
	Exogenously determined variables:
S	Number of stores that may be queried
A	Number of attributes that a product possess
a_{ij}	Attribute of the jth product at the ith store
	Shopbot decision variables:
Q, R, P	Total number of queries to be started, retrievals completed, and presentations to show the consumer, respectively
$\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{p}$	S x 1 vectors of indicator values to determine which sites to query, retrievals completed, and presentations to show the consumer, respectively
t^*	Time for the shopbot server to wait before terminating the Q query threads
	Random variables:
W	Time to retrieve set of R offers
$\boldsymbol{T}$, $\boldsymbol{t}$	S x 1 random vector of random and realized retrieval times for each store
T_i, t_i	Random and realized time associated with retrieving an offer from the ith store
$\boldsymbol{U}$	S x 1 vector of random utility associated with each alternative inclusive of all costs
U_i	Total random utility associated of the ith alternative inclusive of all costs
$\mathring{U}_i$	Utility derived directly from the product, not including waiting time or other costs
$\bar{U}_i$	Component of utility determined by product attributes
ϵ_i	Random utility error of the ith alternative
	Utility parameters:
β_{ij}	Utility associated with the jth attribute of the ith product
ξ	Disutility from waiting one second
ω	Disutility from waiting due associated with shopbot server overhead from threads
λ	Disutility associated with comparing an attribute pair
	Distributional Parameters:
θ	Scale parameter for extreme value distribution
ψ	Exponential parameter
μ	Mean of logistic distribution
σ	Scale parameter of logistic distribution

Table 1. A list of variables and their definitions.

3.3 Deciding which retrieved offers to present to the consumer

We begin by considering the shopbot's decision at the final stage where the shopbot must decide which offers to present to the consumer by choosing $\boldsymbol{p}$, which implicitly defines P. At this stage the shopbot has already decided which stores to query (this decision will be considered in §3.4) and retrieved a set of R offers ($\boldsymbol{r}$). Since the offers have been retrieved the product attributes (price, delivery cost, etc.) are known at this point, hence the random variate representing utility, $\bar{U}_i$, will take the observed value $\bar{u}_i$, where $i \in \ll \boldsymbol{r} \gg$. There is still uncertainty associated with the random variates $\mathring{U}_i$ due to the consumer's evaluation error ϵ_i. Using the definitions given in §3.1 the optimization problem that corresponds with this stage given in (7) can be written as:

$$\max_{\boldsymbol{p}} E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid \bar{\boldsymbol{U}}\langle\boldsymbol{r}\rangle] = E[\max(\mathring{\boldsymbol{U}}\langle\boldsymbol{p}\rangle) - \xi W - \omega Q - \lambda(A-1)(P-1) \mid \bar{\boldsymbol{U}}\langle\boldsymbol{r}\rangle] \; . \tag{8}$$

This simplification uses the fact that the disutility terms for waiting time, server overhead, and cognitive costs are functions of the set of alternatives retrieved, hence they can be extracted from the maximization function. The expectation is computed conditional on the information set $\{\bar{U}\langle\boldsymbol{r}\rangle, \xi, \omega, \lambda, \theta\}$, which is suppressed for notational convenience throughout this subsection.

Clearly the best alternatives to present are those with the highest expected utilities. (Note this simplification depends upon the i.i.d. assumption of ϵ_i.) To find the set of $\boldsymbol{p}$ offerings that maximize the expected utility of this set we order the retrieved offers by their expected utilities. The utilities of the alternatives in the offer set is denoted as $\{\bar{u}_{R:R}, \ldots, \bar{u}_{R-P+1:R}\}$, where $\bar{u}_{R:R} \geq \bar{u}_{R-1:R} \geq \cdots \geq \bar{u}_{R-P+1:R}$. This reduces our problem to a decision about how many offers to present (P). Given P the elements of $\boldsymbol{p}$ are determined by the relation: p_i=1 if r_i=1 and $\mathring{U}_i \geq \mathring{U}_{R-P+1:R}$ and p_i=0 otherwise.

The properties of the extreme value distribution imply that the maximum variate, $\mathring{U}=\max(\mathring{U}_{R:R}, \mathring{U}_{R-1:R}, \ldots, \mathring{U}_{R-P+1:R})$, will also follow an extreme value distribution with location of $\theta\ln(\exp\{\bar{u}_{R:R}/\theta\}+\exp\{\bar{u}_{R-1:R}/\theta\}+\cdots+\exp\{\bar{u}_{R-P+1:R}/\theta\})$ and scale of θ. Hence the expected utility of the offer set is:

$$E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid P, \bar{\boldsymbol{U}}\langle\boldsymbol{r}\rangle] = \theta \ln\left(\sum_{i=1}^{P} \exp\{\bar{u}_{R-i+1:R}/\theta\}\right) + \theta\gamma - \xi W - \omega Q - \lambda(A-1)(P-1) \; . \tag{9}$$

The ordering of the items to include is determined by the sorted order of $\bar{u}_{R-i+1:R}$. To determine the number of elements to include in this set, notice that the two terms that involve P, $\theta \ln(\sum\exp\{\bar{u}_{R-i+1:R}/\theta\})$ and $-\lambda(A-1)(P-1)$, are monotonically increasing and monotonically decreasing in P, respectively. Therefore we can find the optimal value for P by first evaluating at P=1 and subsequently incrementing P until the expected utility begins to decline. This yields the following stopping rule to determine the optimal value P^*:

$$E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid P = P^* + 1, \bar{\boldsymbol{U}}\langle\boldsymbol{r}\rangle] < E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid P = P^*, \bar{\boldsymbol{U}}\langle\boldsymbol{r}\rangle] \Rightarrow$$

$$\frac{\exp\{\bar{u}_{R-P^*:R}/\theta\}}{\sum_{i=1}^{P^*} \exp\{\bar{u}_{R-i+1:R}/\theta\}} < \exp\left\{\frac{\lambda(A-1)}{\theta}\right\} - 1 . \qquad (10)$$

In other words, find the largest P such that the relative gain from adding this alternative exceeds the added cognitive costs to the consumer of its evaluation. The vector of offers to present ($\boldsymbol{p}$) is implicitly defined by setting those elements that correspond with the indices of $\{\bar{u}_{R:R}, \ldots, \bar{u}_{R-P^*:R}\}$ to unity and zero otherwise.

A special case with identical offers: To proceed further we assume that all offers have the same value, $\bar{u}_{R-i+1:R}=\bar{u}$. Therefore the question is not which offers to present (since they are identical), but how many. Under this assumption we can simplify (9) as:

$$E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid P, \bar{u}_i = \bar{u}] = \bar{u} + \theta(\gamma + \log(P)) - \xi W - \omega Q - \lambda(A-1)(P-1) \qquad (11)$$

If we allow P to take non-integer values, we can differentiate (10) with respect to P and find the optimal value:

$$P^* = \frac{\theta}{\lambda(A-1)} \qquad (12)$$

Notice that the optimal set size increases as cognitive costs decrease ($\lambda \to 0_-$) or the variance (θ) of utility increases. Additionally, we can then show that (11) creates an upper bound of (9) by setting $\bar{u}$ to the maximum value, $\bar{u}_{R:R}$, and that (12) bounds the solution in (10).

Example: To illustrate these relationships suppose the average book generates 10 utils with a standard deviation of about 2 utils, i.e., $\bar{u}$=9.1 and θ=1.6. Additionally, assume that a book has 4 attributes (brand, price, shipping cost, and delivery time), i.e., A=4, and set ξ and ω to zero without loss of generality in this example. This means P=1.6/3λ. Suppose each additional unit of cognitive effort decreases utility by .1 utils (λ=.1). The optimal number of offers to present to the customer is 5.3, or rounding our shopbot should present 5 books. Doubling the effects associated with cognitive effort (λ=.2) will reduce the set to P^*=2.7 or about 3 books. Again suppose λ=.1 then the utility generated by offering five books (the optimal number) is 11.4. If the shopbot were to present 20 books then utility would drop by 20%. In summary, naively presenting all offers retrieved is not optimal. Clearly, cognitive efforts are a crucial component in the design of a shopbot.

3.4 Deciding which stores to query and how long to wait for a response

At this initial stage the shopbot must decide which stores to query, $\boldsymbol{q}$, and whether those queries should continue until completion or whether they should be interrupted prematurely at time t^* as stated in (6). We assume that the shopbot will make the optimal decision about which retrieved offers to present as discussed in §3.3. Neither the retrieval times nor the offers are known as in the previous subsection, hence both are assumed to be stochastic variates. The utility from the product attributes (price, delivery time, availability, etc.) is the random variate $\bar{U}_i$ and not the observed value $\bar{u}_i$ as in the previous subsection, where $i \in \ll \boldsymbol{q} \gg$. We assume that response time and utility are independent, hence discount stores will return responses as quickly as expensive stores. Empirically this assumption will be justified by the discussion in §4. Formally we can make use of the result from the previous section and rewrite (6) as follows:

$$E[\max(\boldsymbol{U}<\boldsymbol{p}>)] = E[E[\max(\boldsymbol{U}<\boldsymbol{p}>) \mid \bar{\boldsymbol{U}}<\boldsymbol{r}>]] \qquad (13)$$

Equation (13) makes use of the following relation: $\mathrm{E}[X]=\mathrm{E}[\mathrm{E}[X \mid Y]]$.

The solution of the inner expectation was given in (9), and we will assume that $\boldsymbol{p}$ is chosen optimally by sorting the retrieved values and using the first P^*, where P^* is a function $\boldsymbol{U}<\boldsymbol{r}>$ as defined in (10). The outer expectation requires integrating over the distribution of product utility and retrieval times and summing over all possible permutations of retrieval sets weighted by their probability:

$$E[E[\max(\boldsymbol{U}<\boldsymbol{p}>) \mid \bar{\boldsymbol{U}}<\boldsymbol{r}>]] =$$

$$\sum_{\boldsymbol{r}\in\mathscr{F}} Pr[\boldsymbol{r}]\, E\left[\theta \ln\left(\sum_{i=1}^{P_r^*} \exp\{\bar{\boldsymbol{U}}<\boldsymbol{r}>_{R_r-i+1:R_r}/\theta\}\right) - \lambda(A-1)(P_r^*-1) \mid \boldsymbol{r}\right] + \theta\gamma - \xi E[W] - \omega Q\,. \qquad (14)$$

Where $\mathscr{F}$ denotes all possible 2^Q permutations of the query set, $\Pr[\boldsymbol{r}]$ denotes the probability that query set $\boldsymbol{r}$ is retrieved, $R_r=\sum r_i=\boldsymbol{r}'\iota$ which is the number of retrievals made from query set $\boldsymbol{r}$, P_r^* denotes a stochastic variate that represents the optimal number of offers to present given the retrieved set $\boldsymbol{r}$, and $\bar{U}\langle\boldsymbol{r}\rangle_{Rr-i+1:Rr}$ denotes the R_{r-i+1} ordered statistic from the set $\bar{U}\langle\boldsymbol{r}\rangle$ with R_r elements. The expectation is computed conditional on the information set $\{\boldsymbol{q}, t^*, \xi, \omega, \lambda, \theta, \beta\}$, which is suppressed for notational convenience throughout this subsection.

Additionally to avoid problems when the rare occurrence of no offers are retrieved, we assume that one inferior offer is always available. In our bookstore example this will be a special order that is twice the list price and takes six months to receive. As long as any offer is retrieved this offer is dominated and will not be considered or presented.

The probability that any member of $\mathscr{F}$ is retrieved can be computed in the following manner. Consider the probability that all items are retrieved, it equals $\Pr[t_{i[1]} \leq t^*, t_{i[2]} \leq t^*, ..., t_{i[Q]} \leq t^*]$, where $i[j]$ denotes the jth element of $\ll\boldsymbol{q}\gg$. The probability that all items except the first one is retrieved is: $\Pr[t_{i[1]} > t^*, t_{i[2]} \leq t^*, ..., t_{i[Q]} \leq t^*]$. The remaining elements can be computed in a similar manner. Notice that $\mathscr{F}$ has 2^Q members and its evaluation leads to a computational problem due to the large number of combinations. A set of 10 stores yields 1,024 combinations to evaluate, while 30 stores will lead to more than one billion combinations to evaluate. Before considering the optimal solution to this problem we discuss distributional assumptions for waiting times and utility.

Waiting times are exponentially distributed

Equation (14) requires the computation of $E[W]$, to proceed further we assume that retrieval times follow an exponential distribution and are independently and identically distributed. This assumption is supported by our empirical analysis of §4.2. The probability of observing a query to a selected store is denoted by $\tau \equiv \tau(t^*) = \Pr[T_i < t^*] = 1 - \exp\{-t^*/\zeta\}$. The expected time to observe a queried store is $\mathrm{E}[T_i] = \zeta$ with variance $\mathrm{Var}[T_i] = \zeta^2$. The expected time to observe the set of queries is $\mathrm{E}[\max(\boldsymbol{T}\langle\boldsymbol{q}\rangle)] = \zeta \sum_j 1/j$ where j goes from 1 to Q. For example, if the expected time to observe one store is 2 seconds, then the expected time to observe a set of ten stores is 5.9 seconds. The density function of the maximum variate is $\Pr_{\max(t\langle q\rangle)}[t] = Q\,\tau(t)^{Q-1} \exp\{-t/\zeta\}/\zeta$. We can derive the expected value of W, which can alternatively be described as the maximum from a distribution of exponential variates censored at t^*:

$$E[W] \;=\; E[\min(\max(\boldsymbol{T}\langle \boldsymbol{q}\rangle),\, t^*)] \;=\; \int_0^{t^*} Q\tau(t)^{Q-1}\exp\{-t/\zeta\}/\zeta\, dt \;+\; t^*(1-\tau(t^*)^Q)\,. \qquad (15)$$

If $t^* <$ E[max($\boldsymbol{t}\langle\boldsymbol{q}\rangle$)] then $E[W] \approx t^*$. The probability of observing an individual member $\boldsymbol{r}$ from set $\mathscr{T}$ is:

$$Pr[r] \;=\; \tau^R(1-\tau)^{Q-R} \text{ where } R = \textstyle\sum_i r_i\,. \qquad (16)$$

Utility is logistically distributed

The expectation in (14) implicitly requires the integration of (9) over the distribution of $\bar{U}\langle\boldsymbol{t}\rangle$. To proceed further we make additional assumptions about the distribution of product utility. The natural choice is to assume that the attributes, such as price, are normally distributed at each store, and therefore utility itself is normally distributed. Our empirical analysis in §4.1 supports this assumption.

The problem is that the distribution of the order statistics from a normal distribution do not yield closed form solutions. A reasonable approximation to the normal distribution is the logistic distribution[5]. If we assume that the attribute component of utility is identically and independently logistically distributed across stores, $\bar{U}_i \sim L(\mu,\sigma^2)$ for i=1, 2, ..., S. The cumulative distribution function of $\bar{U}_i$ is defined as:

$$Pr[\bar{U}_i \le x] = \left[1 + \exp\left\{-\frac{x-\mu}{\sigma}\right\}\right]^{-1}. \qquad (17)$$

The mean and variance of the logistic distribution are $E[\bar{U}_i]=\mu$ and $\mathrm{Var}[\bar{U}_i]=\sigma^2\pi^2/3$. The mean of the maximum variate of a logistic distribution is $E[\max(\bar{U}_{R:R})] = \sigma(\psi(R)+\gamma)+\mu$, where $\psi(x)=\Gamma'(x)/\Gamma(x)$. Balakrishnan (1992) presents a full discussion of the properties of the logistic distribution.

We can now bound the component of the expectation in (14) that involves the logarithmic function:

---

5. The distribution of the logistic distribution has longer tails than the normal and is more closely approximated by a Student-t distribution (Mudholkar and George 1978). Also notice that the variance of the standard logistic is $\pi^2/3$, hence the variance parameter of the normal distribution should be scaled by $\pi/\surd 3$ before comparing it the scale parameter of the logistic distribution. For a discussion of the approximation of the normal and logistic distributions see David (1981, pp. 77-78).

$$E\left[\theta \ln\left(\sum_{i=1}^{P^*} \exp\{\bar{U}_{R-i+1:R}/\theta\}\right) \mid \boldsymbol{r}\right] \geq E[\bar{U}_{R:R} \mid \boldsymbol{r}] = \sigma(\psi(R) + \gamma) + \mu \ . \tag{18}$$

This lower bound can be derived by factoring $\bar{U}_{R:R}$ and showing that the term in the logarithmic function always exceeds unity. Intuitively, we are focusing on the "best" single product, if we could know the consumer's choice with certainty ($\theta \to 0$) then this inequality becomes an equality.

Solution

A general analytical solution to the maximization of expected utility given in (12) with respect to Q and t^* is not known. However, we can derive a reasonable approximation for certain cases. First, we consider the case where response time is not interrupted, but all queries that are launched and allowed to execute until completion. Second, we consider the more complex case where some queries may be prematurely interrupted when their response time is longer than t^*.

Time is bounded (t^* constraint is active): If we make the four following simplifications: 1) any uncompleted query threads are interrupted at time t^*, 2) P^* is replaced with the approximation defined in (12), 3) we use the lower bound of expected utility from equation (18), and 4) we use t^* to approximate W, then the expected utility in equation (14) yields:

$$\begin{aligned} E[\ E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle) \mid P^*, \boldsymbol{U}\langle\boldsymbol{r}\rangle]] \approx \sum_{j=1}^{Q} \binom{Q}{j} \tau^{Q-j}(1-\tau)^j \{\sigma(\psi(j) + \gamma) + \mu\} \\ + (1-\tau)^Q \mu_0 - (\theta - \lambda(A-1)) + \theta\gamma - \xi t^* - \omega Q \ . \end{aligned} \tag{19}$$

Where μ_0 is the utility from the alternative when no offers are made, and we assume that $\mu_0 < \mu$. The assumption that times and utility are i.i.d. is critical in this simplification, since both the expected utility and the probability the set is realized depend only upon the size of the set and not the stores selected. The optimal solution for t^* given Q is:

$$t^* = \zeta \ln\left(\left\{ \frac{\zeta\xi}{Q(-\mu_0 + \mu + \gamma\sigma + \sigma\psi(Q))} \right\}^{-\frac{1}{Q}} \right) \tag{20}$$

The solution for the optimal value of Q can be found by substituting (20) into (19) and enumerating the values of Q beginning with unity until the expected utility begins to decline.

Time is not bounded ($t^* \to \infty$): If we make the following three simplifications: 1) all query threads that are launched are allowed to run to completion then $\tau \to 1$, 2) the value of P^* is approximated using (12), and 3) we approximate $E[W]=E[\max(T_1,...,T_Q)] \approx \zeta \ln(Q)$, then the expected utility found in (12) yields:

$$E[\max(\boldsymbol{U}\langle\boldsymbol{p}\rangle)] \approx E\left[\theta \ln\left(\sum_{i=1}^{P^*} \exp\{\bar{\boldsymbol{U}}_{Q-i+1:Q}/\theta\} \right) \right] - \lambda(A-1)(P^*-1) + \theta\gamma - \xi\zeta\ln(Q) - \omega Q\ . \tag{21}$$

Replacing the expectation of the logarithmic term in (21) using the lower bound in (18) and using the approximation $\psi(x) \approx \ln(x+.5)$ when $x>2$ yields the following solution:

$$Q^* = \frac{-2\zeta\xi + 2\sigma + \omega + \sqrt{8\zeta\xi\omega + (2\sigma - 2\zeta\xi + \omega)^2}}{4\omega} \tag{22}$$

If either ξ or ζ are small relative to σ and $\omega << \sigma$, then $Q^* \approx \sigma/\omega$. In other words the number of sites to query is directly proportional to the variance of utility, and inversely related to the waiting time associated with the computational overhead of starting additional threads.

4 An Empirical Study of Online Book Prices, Store Response Times, and Utility

In this section and the next we consider a simulation of our shopbot model proposed in §3 to data collected at 28 online book stores over the course of six months, August 1999 and January 2000. We analyze the prices from a sample of over 600 books with unique ISBNs. The sample includes two types of books: New York Times bestsellers (once they were listed we continued to collect prices regardless of their status on the list) and a random sample of books in print. Automated agents were constructed to collect data from two major comparison shopping engines and some individual stores. Our agent collected information about unit price, shipping cost, shipping time, and delivery time. We begin by formulating a predictive model of price in §4.1 using this data, present an analysis of store response times in §4.2, discuss the part-worths of the utility function, the disutility of waiting, and cognitive costs in §4.3. The simulation results of utility are presented in §5.

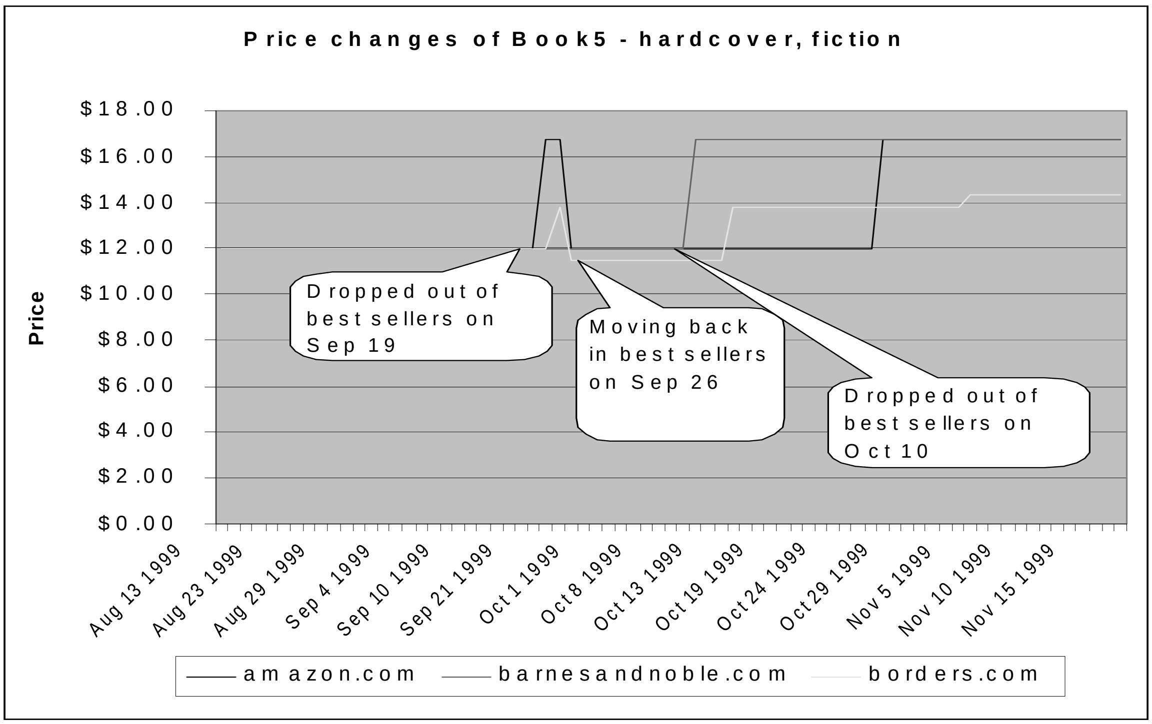

Figure 2. Price changes at the three top online booksellers for a Fiction hardcover book during August through November of 1999.

4.1 Predicting Prices

To illustrate typical price behavior for online bookstores we plot several selected stores in Figure 2. A striking feature of the price series is the persistence of prices. In fact prices may remain at the same level for several weeks. The average time between price changes in our dataset is about four weeks. A practical implication is that the best guess of today's price is yesterday's price. Prices may change for no apparent reason or they may respond to a change in the status of a book on the New York Times bestseller list or a price change at another store. Most stores respond aggressively to a change in the New York Times Bestseller list. If the book is added to the bestseller list there is a high likelihood that prices will drop, and alternately if it drops off the list this may lead to a price increase. However, these effects are not always automatic and there can be delays of several days or weeks before any change results. Additionally, some stores, like buy.com, will respond to changes in prices at another store with high likelihood, while amazon.com seems to act more like a price leader.

These facts taken together suggest that the shopbot can predict prices with a fair degree of precision. The predictability of prices means that shopbots can leverage information from previous retrievals to improve searches by selectively ignoring high priced stores (or stores with low expected utility). We propose a formal statistical model to capture these stylized facts:

$$relprice_{sbt} = \begin{cases} relprice_{s,b,t-1} & \text{with probability } \rho_{sbt} \\ \phi_{s0} + \phi_{s1}\, relprice_{s,b,t-1} + \delta_{sb}' \boldsymbol{x}_{sbt} + \epsilon_{sbt} & \text{otherwise} \end{cases} \tag{23}$$

Where $relprice_{sbt}$ denotes the price at store s for book b on day t relative to its list price ($relprice_{sbt}=price_{sbt}/ listprice_{sb}$). We assume that list prices do not change through time and are known. Relative prices are used to enable direct comparisons across stores and allow the observations to be pooled within bookstores. Notice that our model consists of two components. One states that prices have some probability of changing on each day (ρ_t). If prices are changed, the magnitude of the price change is modeled as an autoregressive transfer function. Also, note that we use the actual price of the product excluding shipping costs and tax. We separately analyzed shipping costs and found them to be the same at a store regardless of the type of book or time the offer was made during our sample. Additionally, we ignore tax. This allows us to simplify the predictive model of price. However, the model could easily be extended to include these values.

Modeling time between price changes: The first component of our model in (23) concerns time between price changes, where time is measured in days (which takes only integer values). Figure 3 that illustrates the days between price changes at 1Bookstreet.com, which is representative of our dataset. The days between price changes ranges between one and 168 days for this store. The median time for a price change to occur is 26.5 days. Most price changes occur fairly infrequently, only one quarter of all prices changed again within the eight days period.

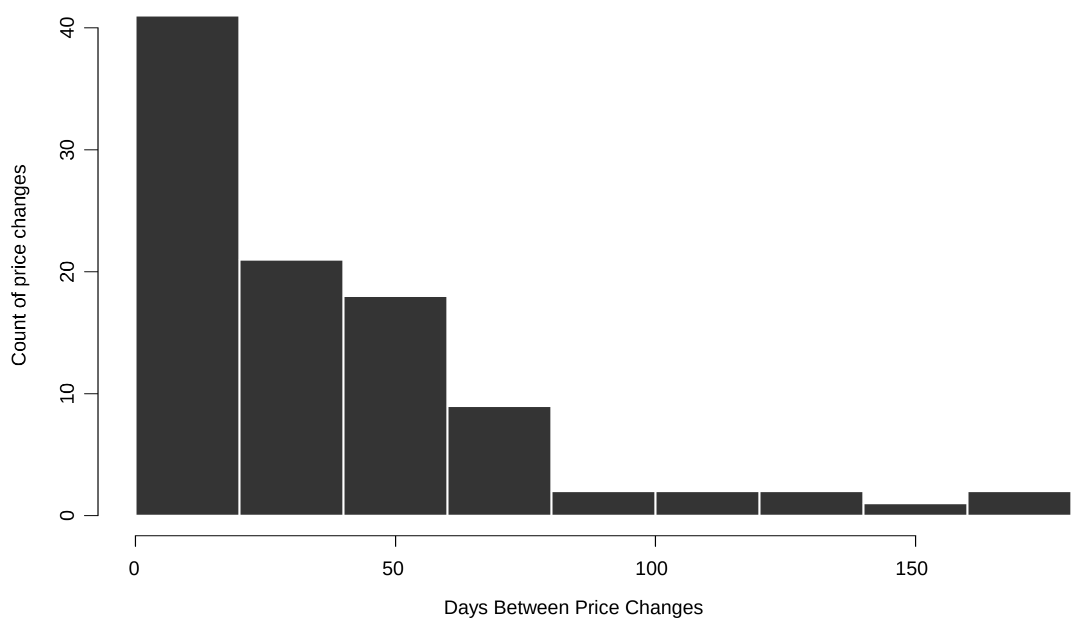

Figure 3. Histogram of days between price changes at 1bookstreet.com. There are a total of 98 price changes that occurred between August 1999 and January 2000 for 40 different books..

A natural suggestion would be to model these counts using a Poisson distribution. Most books in our dataset have only one or two price changes, therefore we pool the data across books within a store. To allow for heterogeneity across books within a store we assume that the Poisson parameter follows a gamma distribution with parameters (γ, δ). The δ parameter is assumed to be constant for all books within a store, while the γ parameter is allowed to vary as a function of covariates, $\gamma_{sbt} = \exp\{\mathbf{x}_{sbt}'\beta_s\}$. This allows us to predict the days to a price change using other information like price changes at other stores or whether the book is on the bestseller list. This assumption results in a negative binomial model for days between a price change (Hausman et al. 1984):

$$Pr[n_{sbt}] = \frac{\Gamma(\gamma_{sbt} + n_{sbt})}{\Gamma(\gamma_{sbt})\Gamma(n_{sbt} + 1)} \left(\frac{\delta}{1+\delta}\right)^{\gamma_{sbt}} (1 + \delta)^{-n_{sbt}} \tag{24}$$

Where this probability defines the probability of a price change used in (23), n_{sbt} is the number of days between price changes. The expected number of days until a price change is $\exp\{\mathbf{x}_{sbt}'\beta_s\}/\delta$ and the variance is $\exp\{\mathbf{x}_{sbt}'\beta_s\}(1+\delta)/\delta^2$.

Store	δ	Constant	Days since change in bestseller status	Number of days since change in price at... 1Book Street	Amazon	Barnes& Noble	buy.com	Borders
1BookStreet	0.04	0.01	0.01		(0.01)	0.00	0.00	0.01
	(0.01)	(0.18)	(0.01)		(0.01)	0.00	0.00	0.00
Amazon	0.06	0.03	0.01	0.01		0.00	0.00	0.00
	(0.01)	(0.15)	0.00	0.00		0.00	0.00	0.00
BarnesandNoble	0.05	0.05	0.00	0.01	(0.01)		0.00	0.01
	(0.01)	(0.15)	(0.01)	0.00	(0.01)		0.00	0.00
Buy.com	0.04	(0.16)	0.01	0.00	0.01	(0.01)		0.00
	(0.01)	(0.13)	0.00	0.00	0.00	0.00		0.00
Borders	0.03	0.02	0.00	0.00	0.00	0.00	0.00	
	(0.01)	(0.15)	(0.01)	(0.01)	(0.01)	(0.01)	0.00	

Table 2. Maximum Likelihood Estimates for time between price changes. Standard errors are given in parentheses below the estimates.

The maximum likelihood estimates of the parameters for the top five online stores are given in Table 2. Notice that days since a change in the bestseller status has a significantly positive impact for amazon and buy.com. This implies that the longer it has been since the bestseller status has changed the less likely a price change, in other words if a price change is to happen it will occur soon after a price change. For most stores the timing of price changes do not appear to be significantly related to price changes at other stores.

New Price conditional on price change: We now consider the second component of the model that determines the magnitude of a price change given that it has occurred:

$$relprice_{sbt} = \alpha + \theta \cdot relprice_{s,b,t-1} + \beta \cdot uph_{sbt} + \lambda \cdot upp_{sbt} + \mu \cdot downh_{sbt} + \gamma \cdot downp_{sbt} + \epsilon_{sbt}, \ \epsilon_{sbt} \sim N(0, \varsigma_s^2) \quad (25)$$

where *uph*, *downh*, *upp*, and *downp* are indicator variables that indicate, respectively, if the book is hardcover and the book moved into bestseller list, if the book is hardcover and the book moved out of the bestseller list, if the book is paperback and book moved into bestseller list, and if the book is paperback and the book moved out of bestseller list.

Store	Constant	Lag Price	uph	downh	upp	uph
1book	0.24	0.07	-0.27	0.17	-0.26	-0.02
	(0.05)	(0.12)	(0.13)	(0.06)	(0.07)	(0.06)
Amazon	0.17	0.69	-0.45	0.23	-0.58	0.43
	(0.04)	(0.08)	(0.03)	(0.04)	(0.03)	(0.04)
Bnoble	0.43	-0.30 .		-0.03	-0.23	0.19
	(0.04)	(0.10) .		(0.05)	(0.07)	(0.05)
Borders	0.24	0.08 .		0.11	-0.30	0.32
	(0.07)	(0.10) .		(0.11)	(0.25)	(0.10)
Buy	0.87	0.01	-0.82	-0.53	-0.74	-0.41
	(0.29)	(0.08)	(1.54)	(0.58)	(1.03)	(0.57)

Table 3. Estimates of the effects on the magnitude of price changes. Standard errors of the estimates are given in parentheses below the estimates.

For the most part when the book moves out of the bestseller list, the price increases, and when it moves back onto the bestseller list the price declines. The magnitude of the effect is larger for hardcover than paperback books. The overall fit of the models is moderate due to the fact that some price changes occur for inventory management issues, periodic price revisions, or other unobserved factors.

Making Price Predictions: To predict price we can join the two conditional components together. Consider the expectation of the one-step ahead, price forecast:

$$E[relprice_{sb,t+1} \mid relprice_{sbt}] = \rho_{sb,t+1}\, relprice_{sbt} + (1 - \rho_{sb,t+1})\left(\phi_{s0} + \phi_{s1}\, relprice_{sbt} + \delta_{sb}' \boldsymbol{x}_{sb,t+1}\right) \qquad (26)$$

This expectation is equal to the probability that a price change has occurred times the conditional expectation plus the probability that a price change has occurred times the probability of no price change. The probability of a price change can be computed from the negative binomial distribution given in (24). Additionally, if $\boldsymbol{x}$ is not known then a forecast of $\boldsymbol{x}$ can be used in its place. Subsequent forecasts can be created in a similar manner by recursively applying this formula. The eventual forecast model under stationarity, or in other words the forecast when no information other than the list price is known follows the usual autoregressive relationship:

$$E[relprice_{sbt}] = \frac{\phi_{s0} + \delta_{sb}' E[\boldsymbol{x}_{sbt}]}{1 - \phi_{s1}} \quad \text{and} \quad Var[relprice_{sbt}] = \frac{\varsigma_s^2}{1 - \phi_{s1}^2} \qquad (27)$$

To illustrate the forecast from this model consider the typical price prediction problem that a shopbot must solve. Suppose that the shopbot performs a weekly search of prices at Amazon on Friday. However, the shopbot may also need to forecast prices during the remainder of the week. The price forecasts under this scenario are illustrated in Figure 4. Notice that on Friday the price and forecast always match, and thereafter the forecast shows a marked affinity for the last price. One mid-week price change is totally missed, while level changes are picked up quickly. If the price change is due to a change in bestseller status then these price changes can be predicted with some degree of confidence.

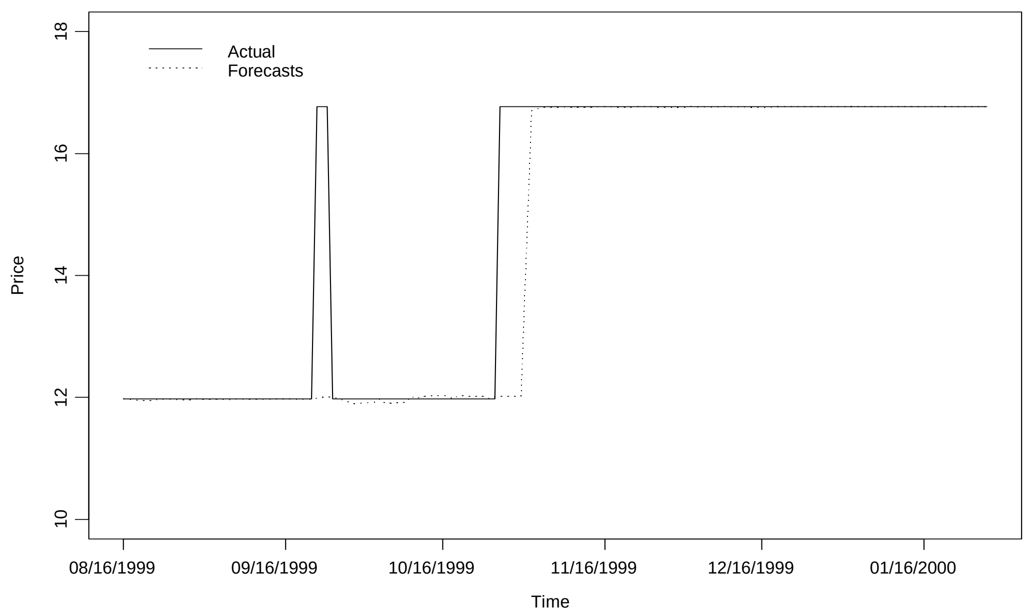

Figure 4. Actual and forecasted price for a selected book at Amazon.

The accuracy of the price predictions depends on the frequency with which the shopbot chooses to gather prices from the stores. The correlations between the actual and predicted price (standardized by its list price) is given in Table 4. Notice if the shopbot has the price for the book from a search three days ago there is a correlation of .99. Clearly, good price forecasts can be generated without having to query the store. As the frequency of price sampling goes down the forecasts start to deteriorate. Although even with month old price information there is a correlation of .82, which implies a shopbot should be able to make an educated guess about the cheapest stores with even fairly old information. Decreasing the frequency of sampling lessens demands on network traffic (and increased demands on an internal database to lookup past prices) but would also lessen the predictive accuracy. This tradeoff between accuracy and speed needs to be considered by a shopbot when choosing an optimal frequency to gather prices.

Price Collection Frequency	Correlation
Only once (Initial Time)	.297
Once every 30 days	.819
Once every 14 days	.914
Once every 7 days	.950
Once every 3 days	.987
Once every day	1.000

Table 4. Correlation between actual and predicted prices for different frequencies of data collection.

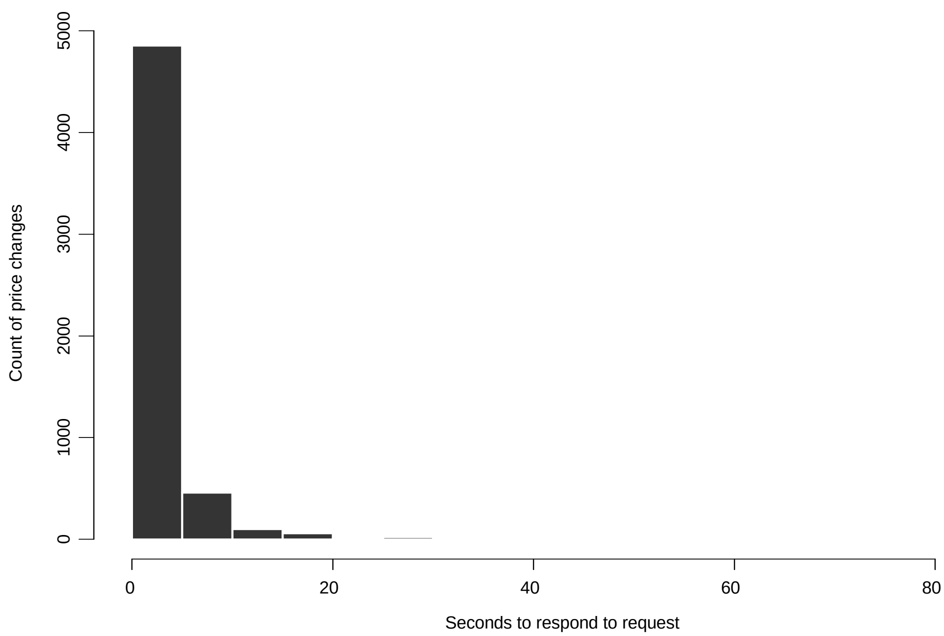

Figure 5. Seconds for Amazon.com to respond to a search request. This data was collected in response to over 5,700 queries between April 2000 and July 2000 between midnight and 6am.

4.2 An Analysis of Store Response Times

Stores typically respond quickly to requests from a user (or shopbot). The time for Amazon's server to respond to 5,700 queries during April to July 2000 are given in Figure 5. For the most part responses are quick with almost 80% of requests being retrieved in less than 2 seconds. However, at certain times network congestion or server overload can cause substantial delays or even no response. In fact in the remaining 20% of requests Amazon.com could take up to 90 seconds to respond. Additionally, about 4% of requests are not returned within a 180 second timeout period. The response times at BCY Bookloft and Barnesandnoble.com are similar.

To model response time we assume that they follow a gamma distribution with probability *p* and do not respond with probability 1-*p*. An estimate of the probability of no response and the gamma parameters for three major stores are given in Table 5.

Store	Number of observations	Probability of response	Gamma Parameters		Moments	
			Location (α)	Scale (σ)	Mean	StdDev
BCY Bookloft	7,803	.979	.452	5.61	2.53	3.77
Amazon	5,739	.960	.775	3.47	2.69	3.05
BarnesandNoble	2,224	.950	.443	5.94	2.63	3.95

Table 5. Estimates of gamma distribution and probability of response for selected bookstores.

If we assume that retrieval times across stores are independent, then the distributional assumption allows us to evaluate the time to retrieve not only a single store but at a set of stores. The time to retrieve a set of *Q* stores will be determined by the time for the slowest store to respond. The mean of the maximum response time can be computed directly. To simplify these calculations we ignore the fact that some stores will not respond and assume that response time is identically and identically distributed as a Gamma distribution with location and scale parameters of .5 and 5 respectively. Therefore, the mean and standard deviation of a single response is 2.5 and 3.5 seconds respectively, which is similar to our sample in Table 5. The expected time for five stores to be queried is 6.9 seconds, while a set of ten stores takes 9.5, and 30 stores would take more than 14 seconds. (These numbers are sample estimates using a simulation with 100,000 draws.) Notice that as the number of stores increases the mean goes up proportional to the logarithm. Therefore, a simple retrieval strategy of searching for all stores may not be a good one since the benefits of retrieving an additional store needs to be balanced against the expected benefits of retrieval.

4.3 Calibrating the Utility Model, Disutility of Waiting, and Cognitive Costs

In this subsection we consider the calibration of the parameters associated with our utility model. Our purpose is to choose reasonable values that will be used for illustrative purposes in a simulation study that will be presented in §5. We do not claim that these settings are correct or representative of an average consumer. They are chosen to present reasonable settings for evaluating the effects of different operational strategies on consumer utility.

The parameters for the utility function measure the implicit tradeoffs consumers are willing to make when evaluating a product. These parameters can be estimated directly from previous purchases at the shopbot or through a conjoint task. We use the maximum likelihood estimates reported by Brynjolfsson and Smith (2000). They measured the utility associated with the following attributes about each book: total price, expected number of days until delivery, and an indicator for brand if it is sold by one of the three large booksellers (Amazon, Borders, and Barnes & Noble). Brynjolfsson and Smith (2000) use actual purchase data from a panel from over 20,000 unique visitors during late 1999.

Parameters	Estimates
Price	
Item Price	-.194 (.001)
Shipping Cost	-.368 (.002)
Expected days until delivery	-.019 (.001)
Effect of branded retailers:	
Amazon.com	.477 (.020)
Barnesandnoble.com	.177 (.023)
Borders.com	.266 (.020)

Table 6. Parameter estimates from Brynjolfsson and Smith (2000) of a multinomial logit choice model for consumer purchases at Dealpilot. The standard errors of the estimates are given in parentheses.

One method for interpreting these coefficients is to make relative comparisons. For example, these estimates imply that for every additional day that it takes to have a book delivered the store needs to decrease its price by $.098 (=-.019/-.194) to keep utility unchanged. To properly compensate a consumer for an extra two weeks in delivery time the store would need to decrease the price by about $1.37. Notice that consumers are almost twice as sensitive to a dollar paid for shipping versus for the item price (this implies it is better to charge higher prices and bundle shipping costs with the book price). Also, the value of Amazon's brand name can be imputed from the model parameters to be \$2.46 more (=.477/-.194) than the value of an less well known bookstore. All three major booksellers show substantial brand equity.

Our model postulates a certain amount of disutility due to waiting one second (ω), waiting for the shopbot server to launch a thread due to system congestion (ξ), and the cognitive costs associated with making comparisons in a final offer set (λ). Unfortunately, the data from Brynjolfsson and Smith (2000) does not include waiting times, nor do we have access to any data that would allow us to empirically answer this question. However, previous studies of Internet behavior clearly demonstrate or hypothesize that there is disutility to waiting. Konana et al. (2000) conjecture that there is a direct tradeoff between waiting time and costs. Dallaert and Kahn (1999) show experimentally that waiting can negatively affect evaluations of web sites. Their results also suggest that waiting is not purely a function of time but can be mediated by other factors. However, for simplicity we have assumed that waiting is a simple tradeoff between time and dollars. Johnson et al. (2000a) argue that the more a web site is used, the faster users can use the web site in the future due to improved knowledge about the web site's design. They estimate that after 5 visits to Amazon the user reduces his time costs by almost \$1.50 per session (or about \$.40/minute) versus having to learn a new online bookstore.

Our analytical framework permits arbitrary values for ω and ξ, but to illustrate our technique we choose plausible settings for the simulation in the following section. To being we assume that the value of time to a consumer is \$.01/second. This translates into a yearly wage of about \$70,000. Therefore every additional second of waiting diminishes utility by .002 utils (=\$.01/sec x -.194 util/\$≈-.002 util/sec), so we let ξ=.002. Suppose the overhead for launching an additional thread is 10 milliseconds then a corresponding value for ω is .00002. If the system is at high utilization then launching and servicing an additional thread could take a substantial more amount of time, hence ω could be substantially higher during high utilization. Finally, we assume that a consumer can make a comparison a second, but attaches ten times the value to a second of cognitive effort versus a second of waiting, e.g., λ=.02. For example, suppose we ask a consumer to evaluate a list of 3 items with 4 attributes, this would require (3-1) x (4-1) = 6 comparisons and have an implicit cost of \$.60 (=.02 util/comparison x 6 comparisons ÷ -.194 util/\$). In comparison if the consumer was simply waiting 6 seconds, the implicit cost of time would be $.06.

5 An Empirical Illustration of Optimal Shopbot Design

In this section we consider a simulated example using the data presented in the previous section to show how the design of the shopbot can influence consumer utility. In these simulations we can solve the shopbot's operational problem empirically which allows us to move beyond some of the assumptions that we were forced to make in §3 to derive analytical results. Specifically, we assume prices are normally distributed and not logistically distributed, we compute the order statistics of utility empirically and do not need to rely upon approximations, the offers are no longer assumed to be i.i.d., and the distribution of time to retrieve an offer are gamma distributed and not exponentially distributed. These simulations allow us to assess the probability that a consumer will prefer shopping at their favorite store versus the present shopbot design or versus the optimal shopbot design. Furthermore we perform a couple of simulations to assess the sensitivity of these results to our parameter settings.

The set of 28 online stores that are present in our database and the shipping offers that can be made by each store are given in Table 7. To construct an actual offer the shopbot needs to search the online store, find the price, and add the shipping cost. For example, if the list price of a book is $19.99, and the actual price at 1BookStreet.com is $15.19, then the book could be delivered by USPS Parcel Post with a delivery range of 6-21 days (expected time to deliver is 13.5 days) for a total cost of $15.19 or UPS 2nd Day with an expected delivery in 9 days for a total cost of $27.14. If only these two offers were presented to the consumer, the utility of these offers would follow an extreme value distribution with locations of -3.204 and -7.516, respectively, and a common scale parameter of unity. The probability that a consumer would chose the first offer would be 98.7%. Clearly the first alternative dominates the second, and illustrates why not all offers need to be presented to the consumer.

Offer	Store	Service	Delivery Range	Expected Days to Delivery	Shipping Cost	Offer	Store	Service	Delivery Range	Expected Days to Delivery	Shipping Cost
1	1 Bookstreet.com	USPS Parcel Post	6-21 days	13.5	$0.00	42	booksamillion.com	Standard Ground	N/A	9.5	$3.95
2		UPS Ground	7-13 days	10	$5.95	43		2nd Day Air	N/A	6.5	$7.95
3		Priority Mail	5-10 days	7.5	$6.95	44		Next Day Air	N/A	5.5	$10.95
4		UPS 2nd Day	9 days	9	$11.95	45	booksnow.com	USPS 4th class	N/A	10.5	$3.95
5	A1 Books	UPS Ground	9-11 days	10	$3.95	46		USPS Priority	N/A	5.5	$4.95
6		Ground Shipping	10-15 days	12.5	$4.95	47		USPS Express	N/A	2	$17.95
7		Priority Mail	7-8 days	7.5	$6.49	48	Borders.com	Standard	5-10 days	7.5	$3.90
8		FedEx 2 Day	7 days	7	$7.99	49		2 Day	5 days	5	$7.95
9		UPS 3rd Day Select	8 days	8	$7.99	50		Overnight	4 days	4	$10.95
10		FedEx Standard Overnight	6 days	6	$11.99	51	Buy.com	Standard Shipping	n/a	7	$3.95
11		UPS 2nd Day (Blue)	7 days	7	$11.99	52		Second Day Air	n/a	3.5	$7.95
12		FedEx Priority Overnight	6 days	6	$14.49	53		Next Day Air	n/a	2	$10.95
13	alldirect.com	USPS Book Rate	N/A	20.5	$3.45	54	Cherryvalleybooks	Book Rate	n/a	8	$3.00
14		UPS Ground	N/A	10.5	$7.00	55		Priority Mail	n/a	4.5	$4.00
15		USPS Priority	N/A	7.5	$11.00	56		UPS 2nd Day	n/a	3.5	$9.00
16	AlphaCraze.com	USPS Special Rate	5-15 days	10	$3.50	57		UPS Next Day	n/a	2	$18.00
17		UPS Regular Mail	n/a	6	$3.95	58	Classbook.com	Standard	6-8 days	7	$4.95
18		Express Priority Mail	3-4 days	3.5	$4.94	59		FedEx	2-3 days	2.5	$19.95
19		UPS 2nd Day Air	n/a	3.5	$11.70	60	Codys Books	1-7 days	N/A	6	$4.00
20		UPS Next Day Air	2 days	2	$20.98	61		3 days	N/A	5	$6.00
21	Amazon.com	USPS Priority Mail	5-10 days	7.5	$3.95	62		2 days	N/A	3.5	$8.00
22		Second Day Air	5 days	5	$7.95	63		1 day	N/A	2	$16.00
23		Next Day Air	2 days	2	$10.95	64	computerlibrary.com	N/A	N/A	10	$0.00
24	Baker's Dozen Online	n/a	n/a	7	$4.00	65	Fatbrain.com	UPS Ground	4-8 days	6	$3.95
25	barnesandnoble.com	U.S. Postal Service	5-9 days	7	$3.95	66		Standard	3-4 days	3.5	$4.95
26		Standard Ground	4-7 days	5.5	$3.99	67		Second Day Air	3 days	3	$7.95
27		FedEx Second Day	3-4 days	3.5	$7.95	68		UPS Overnight	2 days	2	$9.95
28		UPS 2nd Day Air	3-4 days	3.5	$7.99	69	hamiltonbook.com	USPS	N/A	12	$3.00
29		FedEx Overnight	2-3 days	2.5	$10.95	70	Kingbooks.com	USPS Book Rate	16 days	16	$2.50
30	BCY Book Loft	USPS 4th Class	n/a	17	$4.00	71		Standard Shipping	5-9 days	7	$3.95
31		UPS Ground	n/a	9	$5.00	72		2nd Day Air	4 days	4	$7.95
32	bigwords.com	USPS	N/A	7	$4.90	73		Next Day Air	3 days	3	$14.95
33		N/A	N/A	5.5	$5.90	74	Page1Book	US Mail - Priority Service	4-8 days	6	$3.95
34		N/A	N/A	3.5	$11.90	75		UPS Ground	N/A	7	$5.75
35	Book Nook Inc	USPS	n/a	6.5	$3.50	76	Rainy Day Books	N/A	N/A	7	$3.95
36	Bookbuyer's Outlet	Standard	n/a	6.5	$4.50	77	Rutherfords	N/A	N/A	7	$3.50
37		Second Day Air	n/a	4.5	$13.95	78	varsitybooks.com	UPS 2nd Day Air	N/A	5	$4.95
38		Next Day Air	n/a	3.5	$18.95	79		UPS Next Day Air	N/A	4	$17.95
39	Books.com	USPS Book Rate	16-45 days	30.5	$3.85	80	WordsWorth	UPS Standard	5-14 days	9.5	$3.90
40		UPS	4-13 days	8.5	$3.95	81		UPS 2nd Day	4-5 days	4.5	$7.90
41		DHL Express	4 days	4	$10.95	82		UPS Next Day	3 days	3	$10.90

Table 7. Listing of delivery options and shipping costs for 28 online stores.

Before the shopbot queries an online store to determine its price, the shopbot can predict the price using the models estimated in the preceding section. Table 8 presents the mean and standard deviation of expected prices under the assumption that no information is known about the store, see (27) for details on its calculation. For example, without any specific information about past book prices, other than say its list price is $19.99, our best guess is that buy.com will be cheapest with an expected price of $10.39 and a standard deviation of $2.00. Notice the cheaper bookstores tend to have higher variance than more expensive stores.

Store	Mean	Std. Dev.	Store	Mean	Std. Dev.
1bookstreet	0.76	0.13	Borders.com	0.62	0.13
A1 Books	0.75	0.06	buy.com	0.52	0.10
alldirect.com	0.63	0.05	Cherryvalleybooks	0.89	0.02
AlphaCraze.com	0.64	0.09	Classbook.com	0.96	0.06
Amazon	0.63	0.13	Codys Books	0.99	0.06
Baker's Dozen online	0.99	0.06	computerlibrary.com	0.99	0.06
barnesandnoble.com	0.63	0.13	Fatbrain	0.65	0.15
BCY Book Loft	0.72	0.07	HamiltonBook.com	0.70	0.07
bigwords.com	0.77	0.06	kingbooks.com	0.73	0.04
Book Nook Inc.	0.99	0.05	page1book.com	0.99	0.07
Bookbuyer's Outlet	0.62	0.13	Rainy Day Books	0.89	0.05
Books.com	0.70	0.09	Rutherfords	0.89	0.05
booksamillion.com	0.59	0.12	varsitybooks.com	0.75	0.05
booksnow.com	0.88	0.06	WordsWorth	0.83	0.10

Table 8. The mean and standard deviation of prior price expectations for each store without any previous price information at that store. These estimates are relative to the list price, and the predictions are normally distributed.

We now consider the consumer's utility under three scenarios using the parameters specified in the previous section using the data described in Table 7 and 8. The first scenario is that the shopbot searches all stores and presents all results (this is the current decision rule). The second scenario is that the shopbot knows prices with certainty (we assume that the price of the book is equal to its mean and the list price is $19.99). The third scenario that we consider is the case where prices are not known with certainty, but instead the shopbot assumes a priori prices are normally distributed with the means and standard deviations given in Table 8. In both of the latter two scenarios we assume that the shopbot will select the optimal set of offerings to present to the consumer. To simplify calculations we assume that all stores respond to a query (as opposed to a 95% probability that the store will respond). This is not a strong assumption since all stores have similar probabilities of responding and the probability of no response is independent of the offer returned. Incorporating the probability that the store will reduce the utility of all scenarios. Additionally, we assume that a priori we can determine the optimal order of stores to query by sorting on the expected utility (the Appendix contains a discusses the validity of this assumption). Finally, in our simulation if one offer from a store is retrieved, we assume that all offers from that store are retrieved without any extra delay or cost, since delivery costs are deterministic.

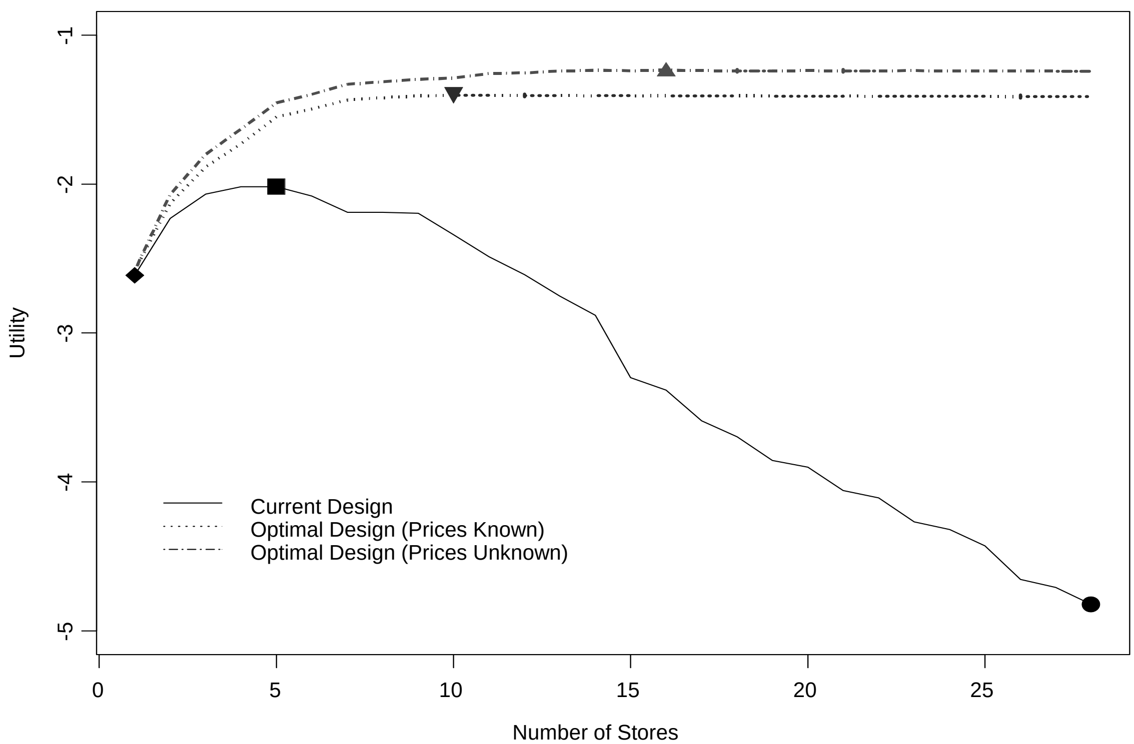

Figure 6 Expected utility based upon the number of stores that are queried using several different shopbot designs.

The expected utility for the three scenarios are plotted in Figure 6 against the number of stores that are to be queried. The stores are ordered according to their best offer. For example, if only one store could be searched then the shopbot would only check 1BookStreet.com. Presumably this would be the consumer's favorite store (denoted by ◆). Notice that the utility for the current shopbot peaks after five stores (denoted by ■) and then starts to decline quickly due to the high cognitive effort placed on the consumer of comparing so many alternatives. In fact, if consumers were asked to choose between the current shopbot design that queries all stores and present all offers (denoted by ●) versus simply visiting their favorite store we would expect that consumers would choose their favorite store by 90%. In contrast, the optimal shopbot designs are not penalized for querying a larger number of stores since they will only select the best offers to present to the consumer. The optimal designs under the assumption that prices are known or normally distributed (denoted by ▼ or ▲) searched ten and sixteen stores, respectively, and would be preferred by consumers over simply visiting their favorite store by 76% and 78%, respectively. Even if current shopbots were scaled back so that they searched fewer stores (but still selected the best stores) consumers would prefer the faster search and smaller decisions sets by 64%.

Offer	Store	Delivery	Service	Price	Shipping	Total
1	1BookStreet.com	USPS Parcel Post	6-21 days	$15.19	$0	$15.19
21	Amazon.com	USPS Priority Mail	5-10 days	$12.59	$3.95	$16.54
51	Buy.com	Standard Shipping	N/A	$10.39	$3.95	$14.34
48	Borders.com	Standard	5-10 days	$12.39	$3.90	$16.29
26	Barnesandnoble.com	Standard Ground	4-7 days	$12.59	$3.99	$16.58
25	Barnesandnoble.com	USPS	5-9 days	$12.59	$3.95	$16.54
42	booksamillion.com	Standard Ground	N/A	$11.79	$3.95	$15.74
16	AlphaCraze.com	USPS Special Rate	5-15 days	$12.79	$3.50	$16.29
64	computerlibrary.com	N/A	N/A	$19.79	$0	$19.79
69	hamiltonbook.com	USPS	N/A	$13.99	$3.00	$16.99
17	AlphaCraze.com	USPS	N/A	$12.79	$3.95	$16.74
70	Kingbooks.com	USPS Book Rate	16 days	$14.59	$2.50	$17.09

Table 9. Listing of offers that will be presented to the consumer for the scenario in which prices are known (and assumed to equal their expected values).

To help understand what offers would be presented, we list the optimal offering set in Table 9 from the optimal shopbot design when prices are assumed to be known (the solution that corresponds to ▼). The ten stores that were queried would yield 32 separate delivery options, but only twelve out of these would be presented to the consumer. We are not simply identifying the cheapest book stores, but those stores that yield the highest utility. For example, Amazon and Borders have higher prices but were included due to the brand equity of their store names and computerlibrary.com and 1bookstreet are included due to their free shipping policies.

Offer	Probability	Store	Delivery	Service	Expected Price	Shipping Cost	Total
1	95.0%	1 Bookstreet.com	USPS Parcel Post	6-21 days	$ 15.19	$ 0.00	$ 15.19
21	80.9%	Amazon.com	USPS Priority Mail	5-10 days	$ 12.59	$ 3.95	$ 16.54
22	1.3%	Amazon.com	Second Day Air	5 days	$ 12.59	$ 7.95	$ 20.54
51	84.3%	Buy.com	Standard Shipping	n/a	$ 10.39	$ 3.95	$ 14.34
52	0.2%	Buy.com	Second Day Air	n/a	$ 10.39	$ 7.95	$ 18.34
48	71.3%	Borders.com	Standard	5-10 days	$ 12.39	$ 3.90	$ 16.29
59	0.5%	Borders.com	2 Day	5 days	$ 12.39	$ 7.95	$ 20.34
25	61.2%	barnesandnoble.com	U.S. Postal Service	5-9 days	$ 12.59	$ 3.95	$ 16.54
26	64.2%	barnesandnoble.com	Standard Ground	4-7 days	$ 12.59	$ 3.99	$ 16.58
27	0.1%	barnesandnoble.com	FedEx Second Day	3-4 days	$ 12.59	$ 7.95	$ 20.54
28	0.0%	barnesandnoble.com	UPS 2nd Day Air	3-4 days	$ 12.59	$ 7.99	$ 20.58
42	56.4%	booksamillion.com	Standard Ground	n/a	$ 11.79	$ 3.95	$ 15.74
43	0.1%	booksamillion.com	2nd Day Air	n/a	$ 11.79	$ 7.95	$ 19.74
16	56.4%	AlphaCraze.com	USPS Special Rate	5-15 days	$ 12.79	$ 3.50	$ 16.29
17	42.7%	AlphaCraze.com	UPS Regular Mail	n/a	$ 12.79	$ 3.95	$ 16.74
18	12.2%	AlphaCraze.com	Express Priority Mail	3-4 days	$ 12.79	$ 4.94	$ 17.73
64	44.6%	computerlibrary.com	N/A	n/a	$ 19.79	$ 0.00	$ 19.79
69	43.3%	hamiltonbook.com	USPS	n/a	$ 13.99	$ 3.00	$ 16.99
70	38.4%	Kingbooks.com	USPS Book Rate	16 days	$ 14.59	$ 2.50	$ 17.09
71	1.2%	Kingbooks.com	Standard Shipping	5-9 days	$ 14.59	$ 3.95	$ 18.54
65	44.2%	Fatbrain.com	UPS Ground	4-8 days	$ 12.99	$ 3.95	$ 16.94
66	23.7%	Fatbrain.com	Standard	3-4 days	$ 12.99	$ 4.95	$ 17.94
67	0.1%	Fatbrain.com	Second Day Air	3 days	$ 12.99	$ 7.95	$ 20.94
13	31.7%	alldirect.com	USPS Book Rate	n/a	$ 12.59	$ 3.45	$ 16.04
36	35.4%	Bookbuyer's Outlet	Standard	n/a	$ 12.39	$ 4.50	$ 16.89
39	2.0%	Books.com	USPS Book Rate	16-45 days	$ 13.99	$ 3.85	$ 17.84
40	17.3%	Books.com	UPS	4-13 days	$ 13.99	$ 3.95	$ 17.94
5	1.5%	A1 Books	UPS Ground	9-11 days	$ 14.99	$ 3.95	$ 18.94
30	2.1%	BCY Book Loft	USPS 4th Class	n/a	$ 14.39	$ 4.00	$ 18.39

Table 10. Listing of best offers and the probability that will be presented to the consumer for the scenario in which prices are assumed to be normally distributed.

If prices are not known with certainty then it is better to search at a larger number of stores. Consider the solution given in Table 10 that corresponds with the case where prices are not known with certainty but are a priori assumed to be normally distributed with the mean and standard deviation given in Table 8 (this solution is denoted by ▲ in Figure 6). This simulation shows that it is best to search at sixteen stores which would yield

55 possible offers. Since prices are not known with certainty the offer set cannot be determined until after the prices are realized, therefore we also list the probability that a offer would be presented to the consumer in the final offer set. Even though a larger number of offers may be potentially included, on average we would expect to only see nine or ten offers presented to the consumer. The ability to select a smaller number of offers demonstrates an important reason why the optimal shopbot design performs so much better than the present shopbot.

The parameter settings play an important role in determining the benefit of the improved shopbot. First consider the case where ω, ξ, and λ are scaled by a factor of 10 to reflect that time is more valuable. If prices are unknown the shopbot would search no more than five stores and most likely only present the best offer retrieved. Again the high cost of time means that the shopbot needs to be much more intelligent in anticipating the tastes of the consumer. In contrast, if time is less valuable, being scale by a factor of .1, then the current shopbot design performs more comparably to the optimal shopbot design. Finally, consider the case where a consumer is indifferent between expending time in a cognitively taxing activity (such as comparing results) and simply waiting ($\xi=\lambda=.002$). Under this assumption the current shopbot design performs more comparably to the optimal shopbot design and the shopbot is more likely to be preferred by the consumer than simply visiting their favorite store.

6 Optimizing Shopbot Profitability

Our discussion up to this point has focused upon optimal shopbot design from a consumer standpoint. However, the pertinent question for shopbot management is how do these operational decisions effect the shopbot's profitability? In this section we consider the connection between utility and profits. The results of the previous section show that there are large utility gains to the consumer of using an optimal shopbot design, although these are diminishing returns. If we temper the expected gains to the consumer with the expected costs to the shopbot of additional search then the consumer's utility maximization solution may not be the same as the shopbot's profit maximization problem.

The total profits (Π) earned by the shopbot during the period of interest (such as a year or quarter) can be decomposed into the profits earned from N consumers who each make M visits to the shopbot. The probability of a purchase on a particular visit depends upon whether utility (V_{ij}) for the ith consumer on their jth visit exceeds some reservation level (R_{ij}), where utility is defined in §3, i.e., V=max($\boldsymbol{U}$<$\boldsymbol{p}$>). If the consumer makes a purchase then the marginal profit earned by the shopbot is π_{ij}. For example, the shopbot may earn a commission upon the sale of a book from the retailer. The fixed costs incurred by the shopbot during this time period is F. We can compute expected profits as follows:

$$E[\Pi] = \sum_{i=1}^{N} \sum_{j=1}^{M_i} Pr(V_{ij} > R_{ij})\, \pi_{ij} - F \tag{28}$$

To complete this specification we define the server load or responsiveness (ω) as endogenous to the system. For example, suppose that the interarrival time between visits is exponentially distributed with a mean of Z seconds, the average time to process a request is t seconds, and server can execute L requests per second. This is a queuing theory problem, that allows us to treat ω endogenously.

Finally, we mention the need to incorporate heterogeneity in consumer response into these calculations. Some consumers may value fast delivery highly, while others think more highly of one store than another. This heterogeneity can be represented by assuming that the β in the utility function follows a multivariate distribution, $\beta \sim N(\boldsymbol{\beta}, V_\beta)$.

A solution to the optimal shopbot design requires data on each of these elements which is not available to us. However, it is straightforward to show that if utility increases, ceteris peribus, then profitability will increase. If the current decision rule of shopbots is to search all stores and present all results, then the results of the previous section show that utility and hence profitability can be increased by querying and presenting a subset of stores appropriately chosen. In other words, our optimal shopbot design requires fewer computational resources than current shopbot design but increases utility to consumers, which implies that our optimal shopbot design is more profitable than the current design of searching all stores and presenting all results.

7 Discussion and Conclusions

We began this article with the observation that shopbots have not fulfilled their original promise of enabling consumers to comparison shop. Our model has provided several insights into how improved shopbot design could increase their value and subsequent use by consumers. Specifically, shopbot design can be improved by selectively presenting and querying stores. Our empirical analysis shows that book prices at online stores can be predicted with a high degree of accuracy without having to query a store but instead rely upon past prices. Embedded within our framework is a compensatory utility model that aids the shopbot in understanding user preferences. This utility model helps the shopbot understand the expected gains to the consumer from more search and balances them against the cost of searching and presenting too much information to consumers.

Our approach to solving the shopbot design problem has taken design elements from computer science and statistics with models of consumer behavior from economics and marketing. This research represents a cross-disciplinary approach that we believe is necessary in the emerging area of research in e-commerce. We believe a dominant research theme in this area is to use models of consumer behavior to better improve the design of web sites. We believe this to be a fertile area for new research. In conclusion, we mention several ideas about future research directions that could improve upon the limitations of our present framework.

1. We have assumed that user preferences are known. However, a shopbot may need to learn from past visits and better anticipate a user's preference function. Most likely consumers will exhibit heterogeneous behaviors, some consumers may be price sensitive and others delivery sensitive. To further complicate matters these preferences may change through time. Consider a student who needs to order a book. If the book is needed for a class that is about to begin next week, the student may be delivery sensitive, while the following week the student may return to their normally price sensitive mind set.
2. An even more sophisticated shopbot could actively query a consumer to determine their preferences for specific stores, delivery times, and prices before or as the query process is carried out. A number of shopbots do query consumers about their preferences before launching a search, but these queries are currently quite crude and laborious. Our suggestion is that the shopbot adaptively ask questions based

upon the information it has about the consumer.

3. An improved understanding of how consumers perceive waiting time is needed. We have assumed a simple framework in which disutility from waiting is proportional to the time spent waiting. However, filler tasks could be performed that could alter consumers perceptions of the time spent waiting. These filler tasks could be used to actively collect information related to the query or could be totally unrelated and simply meant to occupy the user while the search is proceeding.

4. Consumers could choose baskets of items instead of a single item. These baskets can be comprised of both complementary and substitutable products, instead of perfectly substitutable goods as considered in this paper. Consider the purchase of a vacation. It may require the purchase of airline tickets, hotel reservations, an rental car, entertainment, etc. A more advanced shopbot could consider the selection of not only a single product, but the bundle of products.

5. We have not explicitly modeled the shopbot profit function, but instead focused upon one of its input components, consumer utility. However, profits may not be directly proportional to utility. The shopbot may engage in satisficing behavior or need to consider if a consumer will even make a purchase. Additionally, there are many questions about competition between stores and shopbots that also need to be considered.

6. Our shopbot design could also be applied to information goods. Our application to online bookstores was largely due to their popularity and the availability of data. More generically these techniques could be used in text filters, search engines, and recommender systems where decisions about which items to retrieve and present to a user must be made. A common design element is the need to predict the value or utility of an item and to balance the speed of the query with the cognitive demands that will be placed on the user to evaluate the choices.

7. Shopbots could be more proactive in aiding consumers. For example, they could automatically be trained on consumer utility functions and recommend products without being prodded by the user. Additionally, the shopbot may be able to anticipate future price changes and recommend that the user

wait in hopes of finding a better price. Book prices are relatively stable through time, but consider airline ticket prices. A shopbot that could anticipate the likelihood of future price changes might be very valuable, especially for the uninformed consumer.

Appendix

To demonstrate how independence and identical assumptions about $\bar{U}_i$ limits our solution consider the following low-dimensional case. Suppose there are three stores and the shopbot must select the best set of two stores to query. Additionally, we assume utility follows a multivariate normal distribution across the stores and all queries are presented. In notational form these assumptions are S=3, P^*=2, $\boldsymbol{p}$=$\boldsymbol{r}$, and $[\bar{U}_1, \bar{U}_2, \bar{U}_3]' \sim N(\mu, \Sigma)$.

The shopbots problem is to determine which set ({1,2}, {1,3}, or {2,3}) yields the highest expected utility. If X and Y follow a bivariate normal distribution (E[X]=μ_X, E[Y]=μ_Y, Var[X]=σ_X^2, Var[X]=σ_Y^2, and Corr[X,Y]=ρ), Clark (1961) showed that $E[\max(X,Y)]=\mu_X\Phi(\Delta/\upsilon)+\mu_Y\Phi(-\Delta/\upsilon)+\upsilon\phi(\Delta/\upsilon)$, where $\Delta=\mu_X-\mu_Y$ and $\upsilon^2=\sigma_X^2+\sigma_Y^2-2\rho\sigma_X\sigma_Y$. Suppose, that the mean and variance for the three stores are (1,1), $(0,\sigma^2)$, and $(0,\sigma^2)$, where σ>1. In other words, there is a low priced store and two stores with higher expected prices but greater variance. At first glance it would appear that the shopbot should always search at the lower price store. However, if σ>3.67 then the optimal decision is to choose the set with the two higher priced stores {2,3}. The intuition is that the high variance results in a high likelihood of finding a bargain, which compensates for the lower expected value. A positive correlation between stores 2 and 3 would lessen this effect, while a negative correlation would magnify this effect.

This example illustrates that assumptions about independence and identical distributions can alter the decision set of which stores to query. However, the variance has to be high to counter our intuition that we should always search at the lowest priced stores. In most circumstances we expect searching at the stores with the lowest expected prices will be optimal, therefore the real problem is determining how many stores to query.

References

Balakrishnan, N., (ed.) (1992). *Handbook of the Logistic Distribution*, New York: Dekker.

Brynjolfsson, Erik and Michael D. Smith (2000), "The Great Equalizer? Consumer Choice Behavior at Internet Shopbots", CMU Working Paper, Heinz School, Pittsburgh, PA.

David, Herbert A. (1981), *Order Statistics*, 2nd edition, New York: John Wiley & Sons.

Clark, Charles E. (1961), "The Greatest of a Finite Set of Random Variables", *Operations Research*, March-April, pp. 145-162.

Clay, Karen B., Ramayya Krishnan and Eric Wolff (2001), "Prices and Price Dispersion on the Web: Evidence from the Online Book Industry", NBER Working Paper 8271.

Dellaert, B. G. C., and B. E. Kahn (1999), "How Tolerable is Delay: Consumers' Evaluations of Internet Web Sites after Waiting", *Journal of Interactive Marketing*, Vol. 13., No. 1 (Winter), pp. 41-54.

Hann, Il-Horn, Lorin M. Hitt, and Eric K. Clemens (2000), "The Nature of Competition in Electronic Markets: An Empirical Investigation of Online Travel Agent Offerings", Working Paper, GSIA, Carnegie Mellon University, Pittsburgh, PA.

Hausman, Jerry, Bronwyn H. Hall, and Zvi Griliches (1984), "Econometric Models for Count Data with an Application to the Patents-R&D Relationship", *Econometrica*, Vol. 52, No. 4 (July), pp. 909-938.

Johnson, Eric J., Steven Bellman, and Gerald Lohse (2000a), "Keeping Customers on the Web: Cognitive Lock In and the Power Law of Practice", Columbia University, Working Paper. Presented at the 7th Behavorial Decision Research in Management Conference at the University of Arizona, Tucson, Arizona.

Johnson, Eric J., Wendy Moe, Peter Fader, Steven Bellman, and Jerry Lohse (2000b), "On the Depth and Dynamics of World Wide Web Shopping Behavior", Columbia University, Working Paper. http://ecom.gsb.columbia.edu/Papers/newdepth.pdf

Konana, Prabhudev, Alok Gupta, and Andrew B. Whinston (2000), "Integrating User Preferences and Real-Time Workload in Information Services", *Information Systems Research*, Vol. 11, No. 2, June 2000, pp. 177-196.

McFadden, D. (1980), "Econometric Models of Probabilistic Choice Among Products", *Journal of Business*, 53, 13-26.

Mudholkar, G.S. and E.O. George (1978), "A remark on the shpae of the logistic distribution", *Biometrika*, 65, 667-668.

Nielsen, Jakob (2000), *Designing Web Usability*, Indianapolis, Indiana: New Riders Publishing.

Shugan, Steven M. (1980), "The Cost of Thinking", *Journal of Consumer Research*, Vol. 7, September, pp. 99-111.