Designing a Better Shopbot

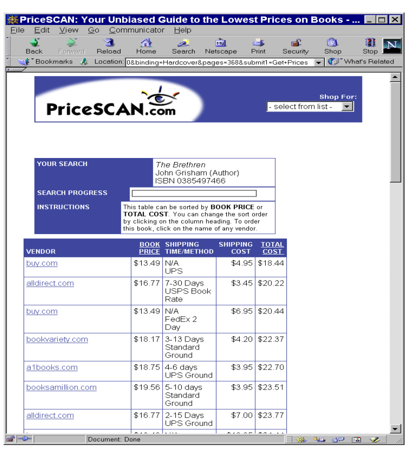
(with Karen Clay, Ramayya Krishnan,
and Kartik Hosanager)

Alan Montgomery
Associate Professor
Carnegie Mellon University
Graduate School of Industrial Administration

e-mail: alan.montgomery@cmu.edu
web: http://www.andrew.cmu.edu/user/alm3

Penn State, eCommerce Workshop, 25 October 2000

What is a shopbot?



Shopping Ro**bot**'s automatically search a large number of stores for a specific product

<u>Example:</u>
John Grisham's *The Brethren*, list price $27.95, prices range between $13.49 (buy.com) and $50.75 (totalinformation.com)

Design involves:
- Computer science (agents)
- Economics (value of price search)
- Marketing (consumer behavior)
- Statistical Modeling (uncertainty)

Outline

- The State of Shopbots
- Improving Shopbot Design
- Modeling Consumer Utility
- Application to Online Bookstores
- Conclusions

The State of Shopbots

Online Retail Shopping

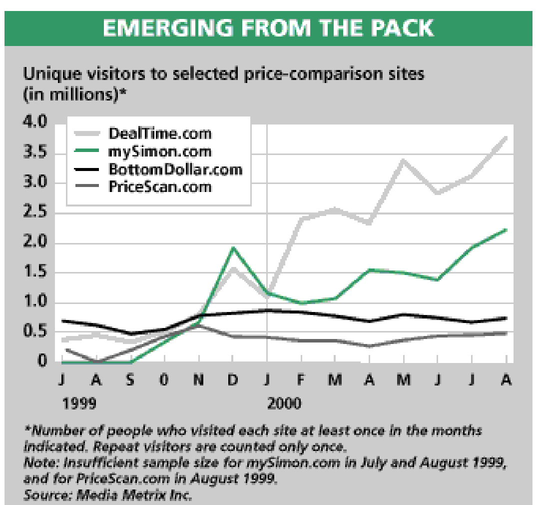
		Millions of Unique Monthly Visitors	% of Web Users
	All Digital Media	**80,097**	**100.0**
	Retail	**53,485**	**66.8**
1	amazon.com	14,464	18.1
2	americangreetings.com	7,719	9.6
3	webstakes.com	5,314	6.6
4	barnesandnoble.com	5,281	6.6
5	mypoints.com	5,269	6.6
6	bizrate.com	5,050	6.3
7	directhit.com	3,952	4.9
8	cdnow.com	3,857	4.8
9	ticketmaster*	3,602	4.5
10	apple.com	3,421	4.3
13	dealtime.com	3,131	3.9
32	mysimon.com	1,915	2.4
	bottomdollar.com	670	0.8

Source:
July 2000

Media Metrix

5

Shopbot Trends

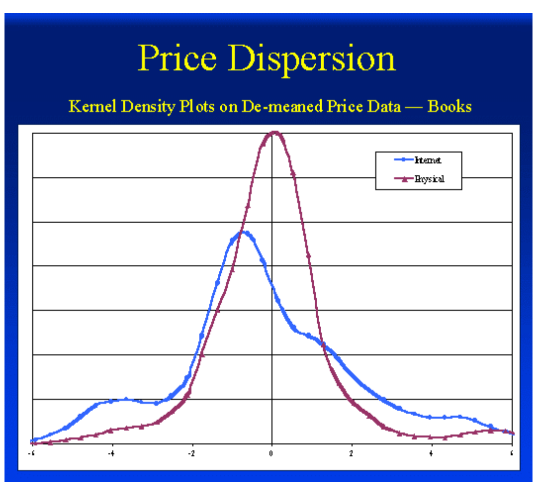


6

Problems with Shopbots

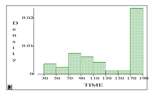
- Less than 10% of shoppers use shopbots, this is up from last year.
- Why don't more people use shopbots?
  - Lack of awareness
  - Lack of benefit (not enough price variation)
  - Lack of information about book (no reviews), you must already know the book
  - Slow response time (the modal time for pricescan and dealpilot is 45 seconds, amazon is <2 seconds)
  - Poor interface, displays too much information

The returns of search



Source: Brynjolfsson and Smith (2000)

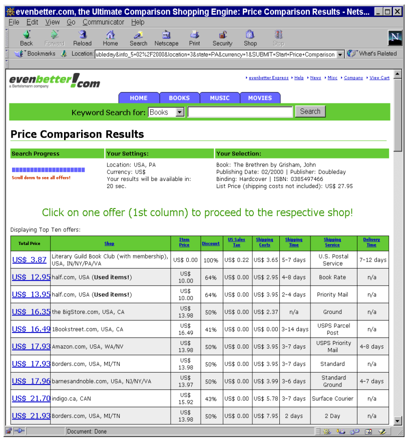
Distribution of Response Times



Response time for evenbetter.com

Improving Shopbot Design

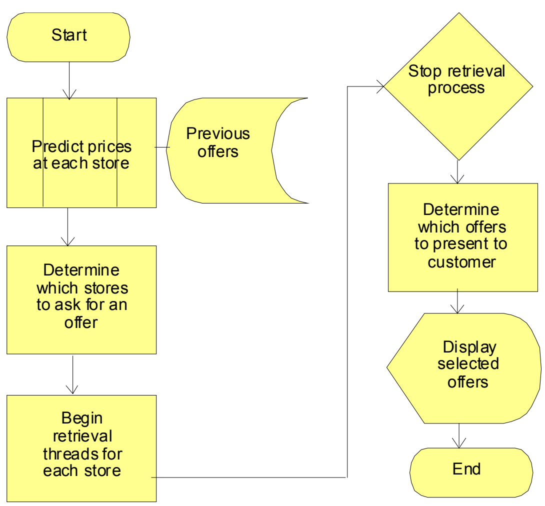
Current ShopBot Design



- Enter a book and evenbetter.com will search 30+ bookstores and return all searches ordered by price
- Yahoo currently lists over 100 bookstores
- Makes search quick and simple
- Search is valuable
  - Average range is $12
  - Amazon lowest only 5% of time (and dropping quickly)

11

Shopbot Operational Flow



12

Operational Decisions

- Which stores to search?
  - Shopbots can form prior expectations about prices that can help eliminate searching at high price stores
- How long to wait?
  - About 5% of store requests time out, also it may be better to interrupt searches at a certain point
- Which offers to present?
  - It is very cognitively taxing for consumers to have to search through scores of offers. Consumer research tells us they will use less efficient comparison rules.

13

Model limitations

- Treat this as a batch job
  - Problem becomes a sequential decision process
- Consumer already knows what they want
  - Imagine designing a shopbot that could find things the consumer did not identify (make tradeoffs in broader product classes)
- We know consumer preferences
  - Allow for random component, but assume that part-worths of utility function are known
- Do not explicitly consider shopbot costs
  - We consider shopbot costs only to the extent they impact waiting time (and therefore utility)
- Do not explicitly consider shopbot profits
  - We presume the shopbot wants to maximize consumer utility $\Rightarrow$ maximize purchase probabilities, however the shopbot may want to lead consumers to purchase specific alternatives

14

Modeling Consumer Utility

Modeling Consumer Interaction with a Shopbot

- Use a compensatory utility model to determine consumer's tradeoff between price, delivery, tax, and waiting time
- Consider the cognitive costs that a consumer incurs in making comparisons
- Use past information from previous web retrievals to intelligently retrieve prices

Utility Model

- Usual additive utility model for the *i*th product given *P* alternatives with *A* attributes in the set:

$$u_i = \overbrace{\sum_{j=1}^{N} \beta_{ij} a_{ij} + \varepsilon_i}^{u_i^*} - \xi \cdot T - \omega \cdot Q - \lambda \cdot C$$

attributes: price, delivery time, etc.

waiting time $=\max(t_1, \ldots, t_M)$

system overhead to process requests

cognitive costs ® (A-1)(P-1)

17

Utility of the Choice Set

- Utility of a set of *M* choices is:

$$\begin{aligned} U &= \max(u_{(1)}, \ldots, u_{(P)}) \\ &= \max(u^*_{(1)} - \xi \cdot T - \omega \cdot Q - \lambda \cdot C, \ldots, u^*_{(P)} - \xi \cdot T - \omega \cdot Q - \lambda \cdot C) \\ &= \max(u^*_{(1)}, \ldots, u^*_{(P)}) - \xi \cdot T - \omega \cdot Q - \lambda \cdot (A-1)(P-1) \end{aligned}$$

present the best offers, ordered observations

18

Formal Problem

- Sequential Optimization – solved backwards

$$\max_{q,p,t^*} \; E[\max(U\langle p \rangle)] \quad s.t. \quad p \le r \le q$$

Variables:

q	offers to query
r	offers retrieved
p	offers presented
*t**	time to interrupt query

19

Example

- Shopbot can search the following stores:
  $Price_1$ ~ N(10,1), $Price_2$ ~ N(11,1), $Price_3$ ~ N(12,1)
- Query stores 1 & 2
  q=[1 1 0], t=[8 12 10]
- Interrupt query at 10 seconds
  t^*=10, **r**=[1 0 0]
- Present offer from store 1 to customer
  p=[1 0 0], U=6

Objective: Maximize utility of the set offered to consumer
Current Solution: q=[1 1 … 1], t*=30, p=[1 1 … 1]

20

Solution

Sequential Optimization Problem:

3. Which offers should be presented, given the retrieval set
2. When should they retrievals be interrupted, given the queries were made
1. Which stores should be queried

3. Which offers to present?

Assume that utility errors follows an extreme value distribution with parameters (0, θ). Usual multinomial logit model. Implies that the maximum also has has an extreme value distribution:

$$\max(u^*_{(1)},\ldots,u^*_{(P)}) = \theta \ln\left(\sum_{i=1}^{P} \exp\{\bar{u}_{R-i+1:R}/\theta\}\right) + \theta\gamma, \quad \bar{u}_i = \sum_{j=1}^{A} \beta_{ij} a_{ij}$$

Which offers to present?

Can now determine expected utility (conditioned on retrieval information set)

$$E[U] = \theta \ln\left(\sum_{i=1}^{P} \exp\left\{ u_{R-i+1:R} / \theta \right\} \right) + \theta\gamma - \xi \cdot T - \omega \cdot Q - \lambda \cdot (A-1)(P-1)$$

Solution:

Start with *P**=*R*

Stop if E[*U*|*P*-1]<E[*U*|*P*]

Let *P*=*P*-1

23

Special Case

Suppose prices are all identical, how many offers to present?

variability (gain to added item)

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

cognitive cost

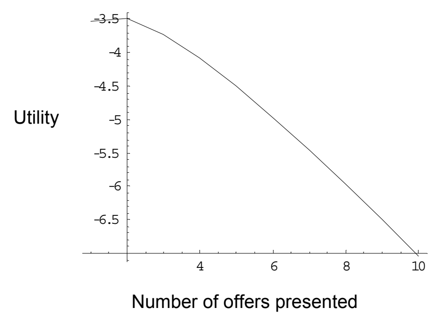
Example

- Average book generates 10 utils with s.d. of 2 utils, i.e., (U=9.1,θ=1.6), Book has 4 attributes (A=4)
- λ=.1 ± *P**=5.3, λ=.2 ± *P**=2.7
- Offer set of 20 books: 8.7 utils
- Offer set of 5 books: 11 utils

24

Implications

Do not show all the results! Utility can decline quickly as a result of added items.



25

2. How long to wait?

Just because a store is queried doesn't imply that it will respond, there is a probability η_i of no response, and let t_i represent time to retrieve offer. If $t_i < t^*$ then observation is censored.

Probability of no response:

$$\tau_i = \eta_i + (1-\eta_i)\Pr\left[t_i > t^*\right]$$

Assume the probability of response independent across stores and also retrieved offer.

26

How long to wait?

Must now evaluate utility over all possible sets of retrievals based on the queries made.

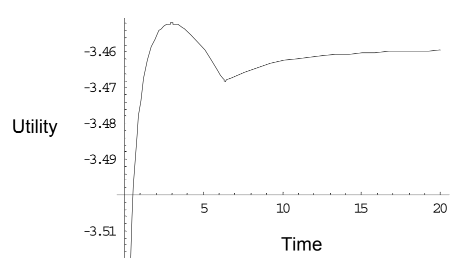
$$\left(\prod_{\iota\in\Omega} \tau_i^{\iota_i}(1-\tau_i)^{\iota_i}\right)\cdot E[U]$$

Ω is set of all possible combinations, dimension is 2^{Q}. Combinatorial explosion, Q=10 yields 1,024 combinations, Q=30 yields one billion.

27

Implications

Suppose times are gamma distribution (mean=1.5, std=2.7), E[max of 10 variates]=7, chance of retrieval=95%.



Act aggressively and truncate response

28

1. Which stores to query?

At this stage the shopbot does not know the price it will retrieve, however it can guess (in fact pretty well).

Order stores based on prior expectations, and the problem now becomes how many stores to query?

29

Example

Suppose there are three stores that may be queried and prices are normally distributed:

$Utility_1 \sim N(1,1)$, $Utility_2 \sim N(0, \sigma^2)$, $Utility_3 \sim N(0, \sigma^2)$

If you could only select two stores which ones will yield the expected maximum utility?

Choose {1,2} if $\sigma<3.67$, otherwise {2,3}

$E[\max(Utility_1, Utility_2)] = \mu_X \Phi(\Delta/\upsilon) + \mu_Y \Phi(-\Delta/\upsilon) + \upsilon\phi(\Delta/\upsilon)$

Where $\Delta = \mu_X - \mu_Y$ and $\upsilon^2 = \sigma_X^2 + \sigma_Y^2 - 2\rho\sigma_X\sigma_Y$

30

Which stores to query?

Unfortunately, while normality may be a good distributional assumption in practice, theoretical properties of its order statistics are not tractable.

A reasonable approximation is to assume that utility (prices) are logistically distributed. Furthermore to yield an analytical result we assume that the stores are i.i.d.. After some work (and approximation) we get the following stopping rule:

$$Q \geq \frac{\sigma}{\omega + \lambda(A-1)(P^*_{R+1} - P^*_R)}$$

Increase set size as the gains to search are higher (σ) and reduce them as the disutility of waiting time increases.

31

Implications

- Suppose there are 10 stores (utility ranges from -4 to -5, std=1)

Larger query size is better, but at a declining rate

32

Application to Online Bookstores

Empirical Components

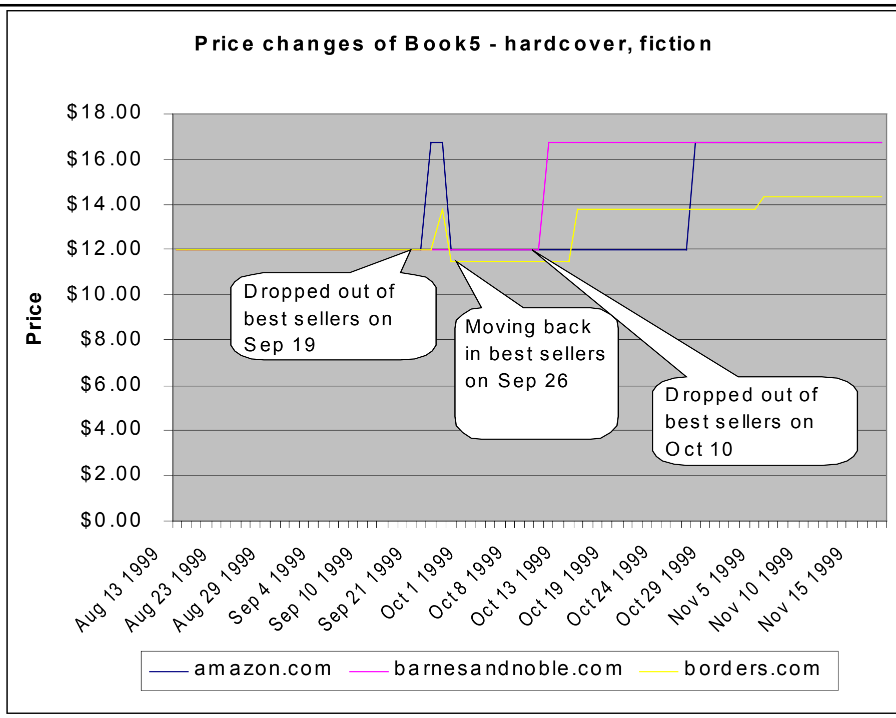
- Predicting Price Changes
- Response Times
- Consumer Utility

34

Data

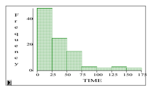
- Automated agents collected data from 2 shopbots and several individual stores
- August 99 - January 00
- 600 books
  - NY Times Bestsellers
  - Randomly selected ISBNs
  - Computer Books

35



Time between price change

Number of days between price changes follows an exponentially declining distribution



37

Predicting Prices

- Predict days between price changes using a Negative Binomial Model with parameters (γ,δ), where:

  $\ln(\gamma) = \alpha_0 + \alpha_0$ days_since_bestseller_change
  $+ \alpha_0$ 1BookStreet $+ \alpha_0$ amazon $+ \alpha_0$ bn
  $+ \alpha_0$ buy.com $+ \alpha_0$ borders

- Given that prices have changed predict the magnitude of price change using an autoregressive model.

  RelPrice(t) = $\beta_0 + \beta_1$ RelPrice(t-1) $+ \beta_2$ uphard
  $+ \beta_3$ uppaper $+ \beta_4$ downhard $+ \beta_5$ downpaper

38

Summary of Results

- Amazon and Barnesandnoble responding quickly to change in bestseller status
- Amazon shows some price leadership, but for the most part weak relationships between price changes at stores
- When books move onto the bestseller list prices drop (more for hardcovers)
- When books move off the bestseller list prices rise

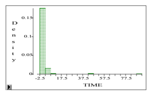
39

Correlation between Actual and Predicted Prices

Price Collection Frequency	Correlation
Only once (initial time)	.30
Once every 30 days	.82
Once every 14 days	.91
Once every 7 days	.95
Once every 3 days	.99
Once every day	1.00

40

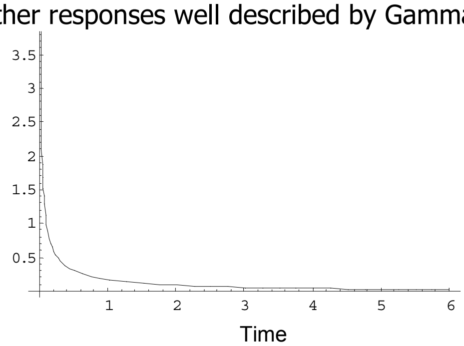
Response Times



41

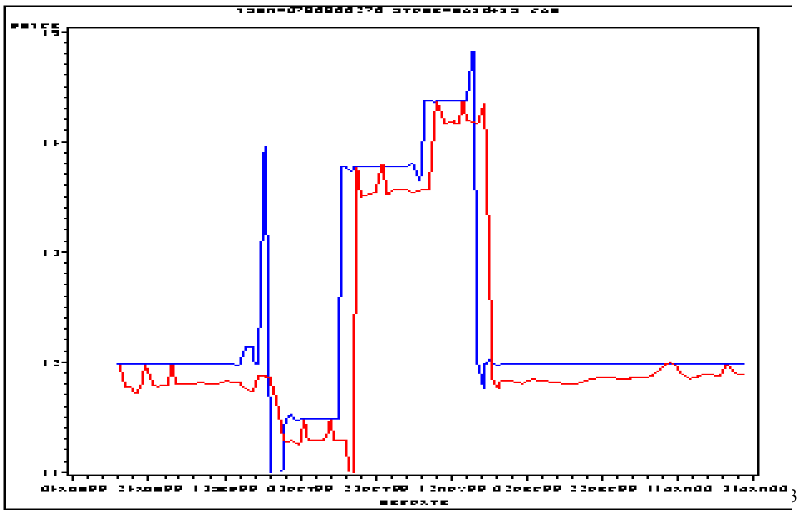
Modeling Response Times

- Mixture of no response and a gamma distribution with parameters (~I=Φ
  - ☐ 95% probability that store will not respond (within 90 seconds)
  - ☐ Other responses well described by Gamma(.3,5)



42

Predicted Price Plot



Shopbot Choice Model

Parameter	Estimate	
Total Price		
Item Price	-.19	($1.00)
Shipping Price	-.37	($1.95)
U.S. Tax	-.43	($2.26)
Delivery Average	-.02	($.10)
Delivery "n/a"	-.37	($1.94)
"Big 3"		
Amazon	.48	($2.52)
BarnesandNoble	.17	($.89)
Borders	.27	($1.42)

44

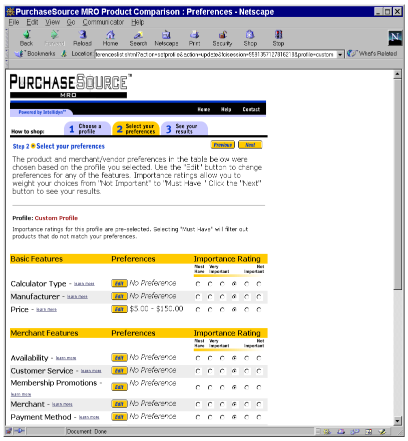
Conclusions

Summary

- Intelligent design of shopbots can dramatically increase the utility that consumers garner from their use
- Instead of passively searching, can incorporate information about utility and price expectations to speed up search and satisfaction
- Incorporates cognitive effort, compensatory utility functions, and information retrieval

46

An improved shopbot?



- Ask users for filtering questions about preferences or use information from previous history
- Appropriately balance the cost of asking for the information with its benefits
- Allow further search
- Better understand how consumers perceive waiting time based on expectations, provide 'filler' tasks

Future Directions

- Optimize on resource consumption at peak traffic hours (using store response time data)
- If thread failures occur, shopbot can make 'best guess' of price at store
- If bestseller status changed today & shopbot knows a store responds to status change in 2 days, it can make recommendations ("wait 2 days and price at amazon likely to be less by $10")
- Personal shopping agent. It knows so much about price, etc. Can complete transactions for users. Useful in a world dominated by pricebots
- How do stores compete effectively? Perhaps use shopbots as a price discrimination tool
- Identify baskets of products or more complex products like travel