Electronic Commerce Research, 3: 277–296 (2003)

Dynamic Consumer Profiling and Tiered Pricing Using Software Agents

PRITHVIRAJ DASGUPTA pdasgupta@mail.unomaha.edu
Department of Computer Science, University of Nebraska, Omaha, NE 68182, USA

P. MICHAEL MELLIAR-SMITH pmms@ece.ucsb.edu
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA

Abstract

Shopbots or software agents that enable comparison shopping of items from different online sellers have become popular for quick and easy shopping among online buyers. Rapid searches and price comparison by shopbots have motivated sellers to use software agents called pricebots to adjust their prices dynamically so that they can maintain a competitive edge in the market. Existing pricebots charge the same price for an item from all of their customers. Online consumers differ in their purchasing preferences and, therefore, a seller's profit can be increased by charging two different prices for the same good from price-insensitive and price-sensitive consumers. In this paper, we present an algorithm that partitions the buyer population into different segments depending on the buyers' purchase criteria and then charges a different price for each segment. Simulation results of our tiered pricing algorithm indicate that sellers' profits are improved by charging different prices to buyers with different purchase criteria. Price wars between sellers that cause regular price fluctuations in the market, are also prevented when all the sellers in the market use a tiered pricing strategy.

Keywords: e-commerce, dynamic pricing, consumer segmentation

Introduction

The Internet and the World Wide Web have been the enabling technologies behind e-commerce. Web sites at which products are displayed and orders are taken have become popular for many types of businesses. Reductions in cost, time and surplus stock result in higher profits to the suppliers. Increased convenience and speed of procurement make e-commerce attractive to buyers. Early e-commerce sites were little more than passive catalogs of products offered by suppliers. The task of finding the best price or the desired product was left for buyers who ventured to do so. The role of an intermediary or a middle-agent in such a model was predicted to be superfluous and the reduction in cost from the intermediary's fee or markup was promoted as an advantage of Web-based shopping.

However, with the rapid growth in e-commerce sites, the task of searching for products from different suppliers has become an onerous burden for buyers. Online search engines [Alta Vista Inc., 1; Google Inc, 13] search millions of pages on the Internet and return an enormous number of results. However, product searches returned by search engines rank results based on factors such as the number of page-hits on Web-sites carrying the product. Attributes such as price or delivery time that are relevant to an online buyer are not easy to extract from those results. Internet intermediaries have reappeared as comparison shopping agents or *shopbots* [22] that can extract relevant product information to the buyer from different online sellers and aid the buyer in arriving at an informed purchase decision. Shopbots can search through millions of Web sites and easily outperform humans in the task of searching and selecting an item online. Sophisticated shopbots reduce the expenditure and improve the satisfaction of online consumers and thus are becoming a popular means of searching for products on the Web.

Comparison shopping enables buyers to be more informed about the offers made by different online sellers for an item. Therefore, an online seller must make an offer that is more attractive than its competitors' offer so that it does not lose its clients to the competitors. This makes it necessary for every online seller to respond rapidly to the selling strategies used by its competitors so that it can continue to make the most attractive offer to customers. Static pricing techniques that employ price forecasts determined from offline processing of sales data is unsuitable in such a rapidly changing environment. In such a scenario, dynamic pricing of items enables an online seller to maintain a competitive edge over its competitors. This makes dynamic pricing for online sellers an important characteristic of Internet markets.

Currently, pricing of goods on most online stores is done statically by humans, possibly with the aid of a software tool [21; 30], that utilizes parameters such as the sale of a good, its production costs, competition between sellers, and consumer trends to predict the price of the good. Online competition makes it necessary for a seller to rapidly change the price it charges so that it can continue to offer the most attractive price in the market. Software agents called *pricebots* provide a suitable paradigm for implementing dynamic pricing algorithms that enable sellers to respond quickly to their competitors' strategies.

Sellers using pricebots that implement dynamic pricing algorithms [Dasgupta and Das, 12; Kephart et al., 18] charge the same price to all the buyers in the market. In such a scenario, when buyers have different purchase criteria, the market is characterized by cyclic price wars between competing sellers. These price wars illustrate the ineffectiveness of the sellers to identify the optimum price that corresponds to the maximum profit. Repeated cycles of price wars also result in less profits to the seller when the price being charged by the seller is in the vicinity of the base cost or production cost of the seller for the good. An Internet based consumer demography study [Clark, 9] shows that the purchasing behavior of online buyers can be broadly classified as price-insensitive and price-sensitive behavior. Buyers who are insensitive to price are willing to spend more money than price-sensitive buyers. Thus, charging a slightly higher price to price-insensitive buyers than price-sensitive buyers yields more profits to a seller. In this paper, we present an algorithm that first classifies the consumer population into price-sensitive and price-insensitive segments, and then determines the optimum price to be charged to each consumer segment that guarantees the highest profit to the seller.

1. Related work

In this section, we provide an overview of the research on online consumer profiling and dynamic pricing. Both these techniques had initially been developed for processing offline sales data. However, Web-based interactions, such as e-commerce, require these models to be adapted for dynamically processing online sales data.

1.1. Consumer profiling for e-commerce

With the popularity of e-commerce a large number of consumers with different preferences and buying criteria have been attracted to shop from online merchants. Online merchants, quite predictably, have responded to the popularity of e-commerce by profiling and segmenting consumers into different personalized user groups [Ben Schafer et al., 4; Caglayan et al., 8; Orwant, 25]. Online merchants such as Amazon [2] have already tested tiered pricing on different types of consumers. Consumer profiling not only ensures increased profits to the sellers by targeting consumer preferences more efficiently, but it also increases consumer satisfaction and ensures return customers.

Consumer profiling or user modeling is not new to market economics [Kay, 16; Orwant, 25; Paiva and Self, 26]. However, with the advent of the Internet, information acquisition and processing have become easier and faster. Whereas earlier techniques primarily relied on offline processing of the gathered data, Internet based consumer profiling techniques also support dynamic processing of the data gathered from consumers to adapt the underlying pricing model of the sellers on the fly. Some of the consumer profiling tools that have been developed over the last few years are:

- **Group Lens** [24] employs collaborative filtering algorithms [Breese et al., 6; Herlocker et al., 15] for predicting users' preferences. These predictions are derived from user ratings based on navigational data such as the number of times a product's Web page is viewed, and the transaction history of a product.
- **Frontmind** [20] uses Bayesian network models to work with incomplete data cases and determine the probability distribution for an attribute or a combination of attributes of the good. The models are also updated dynamically with customer data gathered online.
- **Learn Sesame** [Caglayan et al., 8][1] defines domain models consisting of objects, object attributes and event types. Consumer information is categorized into the domain models. The domain models are then analyzed using incremental clustering to detect consumer behavior and preference patterns.
- **Personalization Server** [Kramer, 19] is based on the offline processing of consumer demographics. It classifies consumers into different groups based on certain rules defined apriori by the seller. The rules are governed by consumer behavior patterns including the consumer's navigation behavior, and the consumer's software, hardware and network environments.

[1] Learn Sesame is no longer available for commercial use.

Most of the consumer modeling software discussed above employ offline processing of data gathered from consumers. This was an acceptable technique when sellers were not empowered with the speed and information provided by the Internet. However, with numerous merchants offering the same good on the Internet, an online seller must adapt to consumer preferences more rapidly to impress and, perhaps entice consumers to shop from its site. In the system presented in this paper, every seller dynamically segments the consumer population and updates those segments while selling its goods. Rapid online updates of consumer behavior enables the sellers to target consumer requests more efficiently.

1.2. Dynamic pricing

Recently online dynamic pricing has stimulated interest in the commercial and research communities. Increased profits and rapidly clearing inventories resulting from efficient pricing has encouraged the development of software pricing tools including Live Exchange [21], Site-sell [30], and Talus Solutions [31].

Several mathematical models have been developed for dynamic pricing. Kephart et al. [18] have viewed the price setting problem as a one-shot game. Sellers employ pricebots and adjust the price that they are charging at certain intervals using different pricing strategies including derivative following, myoptimal pricing, Q-learning and no-regret pricing.

Typically, in determining its own pricing strategy, a seller uses available information about the market, such as the distribution of buyer preferences, or its competitor's prices. There has been recent work in the literature which attempt to address the question of automated dynamic pricing assuming more or less complete information about the market [Greenwald et al., 14; Kephart and Greenwald, 17; Tesauro et al., 32]. But what if the seller has only limited information about its environment? Previous researches have explored how a monopolistic seller might dynamically set its price schedule to maximize profit in a market where it has to learn the buyer preferences [Brooks et al., 7]. In this work, we study markets with multiple sellers competing for the largest market share, where each seller has no information about the buyer preferences or its competitors' prices.

In the traditional economy, obtaining a competitor's pricing information often involves considerable effort, and in certain situations such information may be unavailable (e.g., sealed-bid auctions) or it may be unethical and illegal to gather such data. In contrast, price-checks from competitors in electronic markets is fairly trivial and, indeed, is a common practice. However, there is no guarantee that online competitors will continue to maintain the price they have revealed. An intelligent seller might reset its price slightly at irregular intervals to leave an inquisitive competitor with outdated price information. Moreover, with a huge population of online sellers, price comparison with competitors can become an arduous burden in comparison to the actual task of selling goods.

In [Dasgupta and Das, 11, 12] a similar price setting strategy has been investigated using less information intensive algorithms that calculate the price to be charged by a seller without using information about the competitors' prices and profits. However, in all these algorithms, every seller in the market charges the same price to all of the buyers between two intervals of price adjustments. This encourages sellers to under-cut each other's price to attract as many price-sensitive buyers as they can and capture the largest share of the market. As a result, price-insensitive buyers are charged the price that the price-sensitive buyers are willing to pay. Uniform pricing for all types of consumers gives rise to cycles of price wars among the sellers in the market [Dasgupta and Das, 12; Kephart et al., 18]. These price wars result in very little or no profit to the sellers when operating at a price close to the production cost of the good. By charging the same price to all types of buyers, the seller evidently forfeits the profit it could have made by charging a higher price to the price-insensitive buyers. Therefore, the seller can improve it profits by identifying the price-insensitive buyers and charging them a higher price than the price-sensitive buyers.

In this paper, we present an algorithm for segmenting buyers with different purchasing preferences and then charging a different price for each buyer segment. Consumer segmentation followed by tiered pricing is not new in the field of pricing [Cohen and Ramaswamy, 10; Trivedi, 33]. Telecommunication and Internet service providers employ tiered pricing to attract consumers with different expenditure capabilities. As Batson [3] observes "Tiered pricing allows a new product to gain its normal high introductory price in some affluent markets while providing a marginal, more affordable price to the neediest countries in the world". Our algorithm first partitions the consumer population into different segments determined by their purchase criteria and then charges a different price for each consumer segment.

Conventional consumer modeling software decouples the model from the application by providing the consumer modeling software as separate module from the application software. This has the advantage of increasing portability. However, decoupling affects the performance of the system because some inter-process communication technique must be used to interconnect the application and the consumer modeling software. Rapid consumer modeling is an important aspect of e-commerce. Therefore, the system that we have developed combines the consumer modeling algorithm with the application software that the seller uses to sell the goods on its Web site.

2. Model

We study the price dynamics in the model of the shopbot economy developed by Kephart et al. [18]. As shown in Figure 1, the market consists of S sellers who compete to provide B buyers ($B \gg S$) with a single indivisible commodity. Sellers are assumed to have sufficient supply of the commodity to last the lifetime of the buyers. The goods being sold are consumables and buyers return to the market repeatedly to re-purchase the good. Examples of such markets include commodities used daily such as groceries and even renewable services such as telephone or Internet services. Each buyer has a valuation or reservation price p_{r} corresponding to the maximum unit price it is willing to pay. Buyers are of two types depending on their selection criterion of a seller:

- *A-type* buyers select a seller at random and purchase the good if the price offered by the seller is below the reservation price p_{r}.

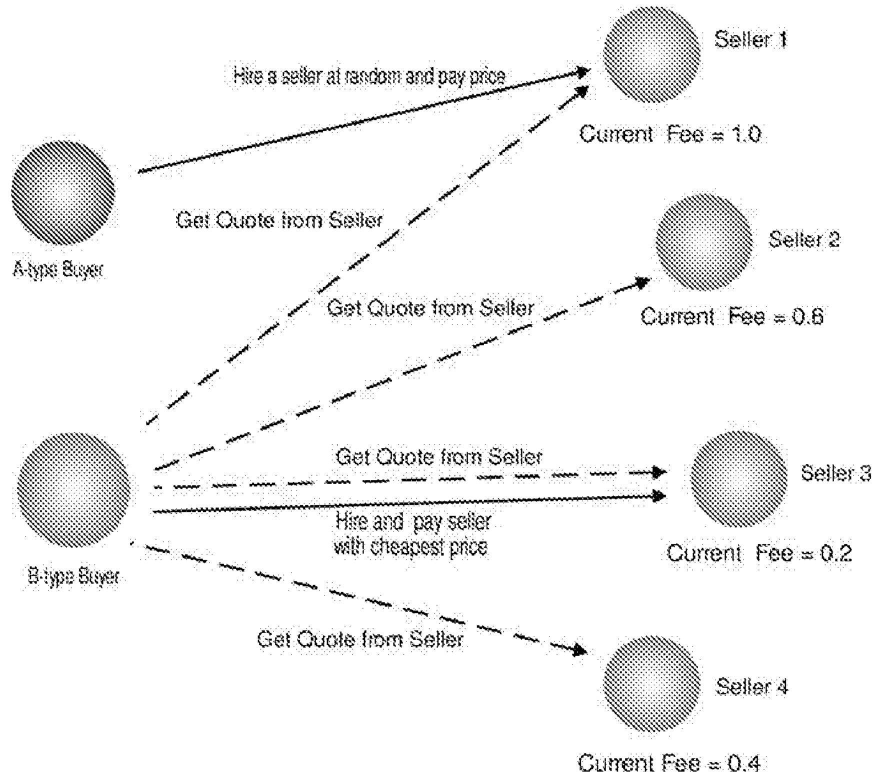


Figure 1. Seller selection criterion of A-type and B-type buyers in the market.

- *B-type* or *bargain-hunting buyers* employ a shopbot to select the seller that offers the lowest price and to purchase the good if the price offered by the seller is below the reservation price p_r. Price ties between multiple sellers are broken randomly.

The ratio of A-type to B-type buyers in the market is assumed to be 25% : 75%, based on a recent study of online consumer types [Clark, 9]. The model also assumes that a buyer's strategy in selecting a seller is uncorrelated with its valuation p_r and that it does not change with time. For our model, we have assumed that the reservation prices for the different buyers are distributed normally.

As shown in Figure 1 buyers enter the market at constant rate ρ_b and request price quotes from sellers. We assume that the buyers are aware of the existence of all the sellers in the market. Since we analyze the price dynamics of sellers, seller discovery is not treated as a major concern in the model. In the real world, online buyers can employ comparison shopping services [22] to discover sellers. After obtaining price quotes, buyers select a seller depending on their type. Buyers return to the market at the rate ρ_b. When a buyer re-enters the market and purchases from a seller from which it has already purchased earlier, the seller tries to identify the buyer's type.

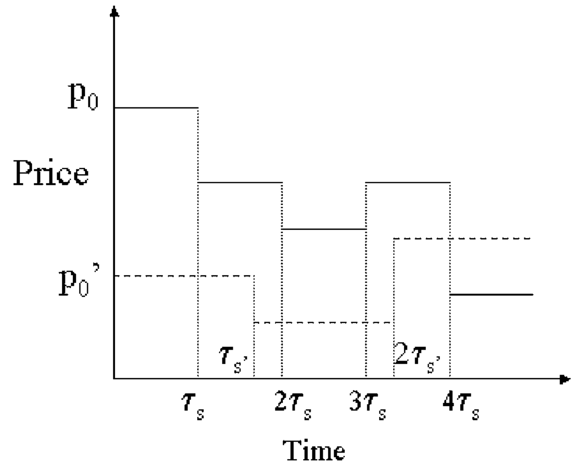


Figure 2. A hypothetical plot of the price vs. time profile of two sellers S and S' with different market entry prices and different price update intervals.

Sellers enter the market with an initial price p_0 for a unit of the good. Sellers are allowed to reset their unit price at intervals τ_s. As each seller adopts its own pricing strategy, the sellers' prices and profits vary dynamically in the market. Figure 2 shows a hypothetical price vs. time plot for two sellers S and S'. S enters the market with a initial price p_0 and adjusts its price at intervals of τ_s, while S' enters the market with an initial price of p_0' and adjusts its price at intervals of τ_s'. The objective of a seller is to set a price at the end of every interval such that its immediate profits during the next interval are maximized. To achieve this, a seller uses the services of a pricebot. In our model, a pricebot uses the model-optimizer (MO) algorithm [Dasgupta and Das, 12] to determine the price at the end of every interval that yields the highest profit under the current market conditions. Other dynamic pricing strategies, including game-theoretic pricing, myoptimal pricing, q-learning and no-regret pricing [Kephart et al., 18], enable a seller using those strategies to respond more rapidly to price fluctuations in the market, as compared to a seller employing the MO algorithm. However, those strategies require information about competitors' prices, competitors' profits, and reservation prices of the buyers. These information are difficult and at times even illegal to obtain over the Internet and, therefore, not suitable for an e-market. Moreover, comparing other seller's prices and profits when different sellers update their prices asynchronously, as illustrated in Figure 2, requires additional synchronization measures that slow down the operation of the seller.

2.1. Model Optimizer algorithm: Non-tiered pricing

The Model Optimizer (MO) algorithm [Dasgupta and Das, 12] utilizes a seller's historical price and profit data to predict a price for the seller to use in successive intervals. A new price that provides the maximum profit in the model is determined during every interval, and this new price is posted to the market. As new price-profit information arrives, the model is continuously updated to reset the posted price.

A major handicap of the above approach is that Internet markets consist of more than one independent seller, and thus, the dynamics of the entire system is non-stationary. In such a situation, it can be argued that a seller has less confidence in applicability of past price-profit relationships in determining its future pricing strategy. This suggests an approach which can assign weights to historical price-profit information in terms of some criterion such as relevance. For this purpose we have used a nonlinear regression approach using least squares. It should be emphasized that the underlying non-stationarity of the system would allow this approach to be applicable only for short-term decision making. Additional information such as consumer preferences and competitors' choices can then be used for more long-term pricing strategies.

Assuming that pricebots have exact information about prices (p) and the measured profits (π) contain all the error or noise, a regression of profit π on price p can be performed. Since π is expected to be non-linearly related to p, we use a high-degree polynomial to model the price-profit data. Let $\pi = c_0 + c_1 p + c_2 p^2 + \cdots + c_r p^r$, where c_i are the coefficients of the polynomial we want to determine. We assume there are m data points (p_t, π_t) each associated with weight w_t. The weights express our confidence in the accuracy or relevance of the points. (Note that m must be greater than r.) The deviation at each point is

$$e_t = c_0 + c_1 p_t + c_2 p_t^2 + \cdots + c_r p_t^r - \pi_t.$$

We now form the weighted sum of squares of the deviations for points (p_t, π_t) with $\tau \leq t \leq \tau + m$

$$F(c_0, c_1, \ldots, c_r) = \sum_{t=\tau}^{\tau+m} \left(c_0 + c_1 p_t + c_2 p_t^2 + \cdots + c_r p_t^r - \pi_t\right)^2 w(t).$$

By setting the partial derivatives of F with respect to the coefficients equal to zero we find the normal equations which can be expressed in the matrix form as

$$\begin{bmatrix} \sum w_t & \sum w_t p_t & \ldots & \sum w_t p_t^r \\ \sum w_t p_t & \sum w_t p_t^2 & \ldots & \sum w_t p_t^{r+1} \\ \vdots & \vdots & & \vdots \\ \sum w_t p_t^r & \sum w_t p_t^{r+1} & \ldots & \sum w_t p_t^{2r} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_r \end{bmatrix} = \begin{bmatrix} \sum w_t \pi_t \\ \sum w_t p_t \pi_t \\ \vdots \\ \sum w_t p_t^r \pi_t \end{bmatrix}.$$

By solving the matrix expression we can find the set of coefficients which best fits a given set of m data points (p_t, π_t) along with their corresponding weights w_t.

After obtaining the polynomial fit on the price-profit relationship, a non-linear optimization scheme must be used to locate the price that corresponds to the maximum profit. We have selected the Nelder–Mead algorithm [Nelder and Mead, 23; Press et al., 27] to identify the price corresponding to the maximum profit in the modeled price-profit relationship. The Nelder–Mead algorithm employs a simplex hill-climbing approach to solve uncon-

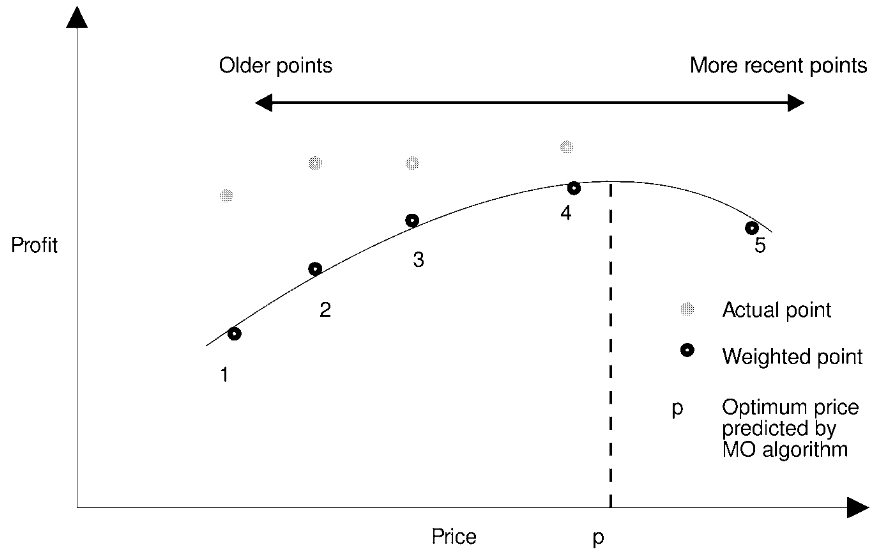


Figure 3. The Model Optimizer (MO) algorithm fits a polynomial to the historic price-profit data of a seller.

strained maximization problems. Our previous work has shown that this approach well suited for price-setting problems in information limited environments [Brooks et al., 7].

Since a seller does not have to depend on competitors' price and profit data before adjusting its own price, the MO pricing strategy is unaffected by the asynchrony in the price adjustments of different sellers in the market. The MO algorithm, as illustrated in Figure 3, works as follows:

– Assign weights to the last n points in the price vs. profit profile of the seller. The weight of a point expresses its relevance to current market conditions. Older points are less relevant and assigned a lower weight, while recent points are more relevant to the current market conditions and are assigned a higher weight. The most recently obtained point on the price-profit profile is not assigned a weight and is taken at its current value.
– Fit a polynomial over the n points in the history window of the seller using a nonlinear regression approach.
– Use a non-linear optimization scheme, like the Nelder–Mead algorithm [Nelder and Mead, 23], to locate the price that corresponds to the maximum profit in the model.

3. Consumer segmentation

A seller using the MO algorithm charges the same price to all of its customers. However, the market comprises two types of buyers, viz., A-type buyers who are insensitive to the price of the good and B-type buyers who select the seller that offers the lowest price for the good. As B-type buyers comprise 75% of the consumers and are predominant in the market, the sellers lower their price to attract more B-type buyers to increase their profit.

However, this uniform pricing for all buyers results in a loss of revenue from the A-type buyers who are willing to pay a much higher price for the good than the B-type buyers. Thus, it makes sense to segment the buyer population into A and B type buyers and charge a different price for each segment.

To segment the consumers, the type of a buyer must be identified from its purchasing behavior. Perhaps, the simplest method for the seller to achieve this is to reveal a slightly higher price than before, when a buyer returns to purchase the same good. If the buyer continues to buy at the higher price, it can be identified as a price insensitive A-type buyer. However, refusing to pay a higher price for the good does not immediately identify the buyer as B-type. A buyer can refuse to purchase from a seller for several reasons, including the seller's price exceeding the buyer's reservation price, an A-type buyer selecting another seller, or a B-type buyer selecting another seller that is offering the cheapest price. This complicates the consumer segmentation problem and, thus, the buyer segments need to be updated continuously as new sales data arrive.

3.1. Consumer segmentation algorithm

To achieve purchase criteria based consumer segmentation, the pricebot of a seller constructs two temporary buyer sets:

- S_{A} containing the buyer ids of the buyers that have have been identified as A-type,
- S_{AB} containing the buyer ids of the buyers whose type has not yet been identified.

The pricebot then charges a high price to the buyers in S_{A} while charging a competitive price given by the MO algorithm for the buyers in S_{AB}.

The parameters used by the consumer segmentation algorithm are:

n_{q}: number of quotes given out by a seller to buyers between two price adjustments,
n_{p}: number of purchases made by buyers from the seller between two price adjustments,
η: purchase rate, $\eta = n_{\mathrm{p}}/n_{\mathrm{q}}$ used to classify buyers as A-type and B-type,
$S_{\mathrm{A},t}$: set comprising only A-type buyers during interval t, $S_{\mathrm{A},0} = \emptyset$,
$S_{\mathrm{AB},t}$: set comprising either A-type or B-type buyers at the end of interval t, $S_{\mathrm{AB},0} = \emptyset$,
$S_{\mathrm{p},t}$: set containing the buyer ids of buyers who purchased during interval t,
$p_{\mathrm{A},t}$: price charged by seller to buyers in S_{A} during interval t,
$p_{\mathrm{AB},t}$: price charged by seller to buyers in S_{AB} during interval t,
n_{a}: actual number of A-type buyers that purchased during interval t at price $p_{\mathrm{A},t}$,
n_{e}: estimated number of A-type buyers that purchased during interval t at price $p_{\mathrm{A},t}$,
p_{co}: production cost of an item for a seller,
$\mu_{\mathrm{r},t}$: pricebot's estimate of the mean of the normal distribution for buyers' reservation prices p_{r} at the end of interval t.

The pricebot calculates the value of η at the end of every interval. If $\eta \simeq 1$, most of the quote requests have matured in sale, thus indicating that the seller is attracting both A-type and B-type buyers. The buyer ids of the buyers that had purchased in the last interval t are then added to S_{AB}. On the other hand, if $\eta \ll 1$, the sales are due only to the few A-type buyers in the market, while the B-type buyers had selected another seller offering a lower price. The buyers that purchased in the last interval t are A-type buyers whose buyer-ids are added to S_{A}. Duplicate buyer ids are removed from S_{AB}. The update rules for the two sets are:

- If $\eta \simeq 1$, then

$$S_{\mathrm{AB},t+1} = S_{\mathrm{AB},t} + S_{\mathrm{p},t}. \tag{1}$$

- If $\eta \ll 1$, then

$$\begin{aligned} S_{\mathrm{A},t+1} &= S_{\mathrm{A},t} + S_{\mathrm{p},t}, \\ S_{\mathrm{AB},t+1} &= S_{\mathrm{AB},t} - S_{\mathrm{A},t}. \end{aligned} \tag{2}$$

3.2. *Tiered pricing for consumer segments*

For the B-type buyers, the pricebot uses the MO algorithm described in Section 2.1 to calculate the competitive price $p_{\mathrm{AB},t}$ at the end of every interval.

The pricing problem for A-type buyers is more complicated than competitive pricing. When a seller charges a very low price from an A-type buyer the seller attracts all A-type buyers. However, the seller also ends up losing a part of its potential profits because the price-insensitive A-type buyer might have purchased the item at a higher price. On the other hand, if the seller attempts to charge a fixed high price to all A-type buyers, then that price might exceed the reservation price for many A-type buyers. Those A-type buyers then reject the seller and once again the seller loses some of its potential profits. Therefore, the objective of every profit maximizing seller should be to determine the optimum price $p_{\mathrm{A},t}$ for every interval t that maximizes the profit:

$$\pi_{\mathrm{A},t} = (p_{\mathrm{A},t} - p_{\mathrm{co}})|S_{\mathrm{p},t} \cap S_{\mathrm{A},t}|, \tag{3}$$

where $|S_{\mathrm{p},t} \cap S_{\mathrm{A},t}|$ represents the number of buyers purchasing during interval t that had already been identified as A-type buyers. If the seller attempts to charge a price higher than $p_{\mathrm{A},t}$ the number of A-type buyers given by $|S_{\mathrm{p},t} \cap S_{\mathrm{A},t}|$ decreases and hence the profit given by $\pi_{\mathrm{A},t}$ also diminishes. Similarly, the profit $\pi_{\mathrm{A},t}$ diminishes if the seller attempts to lower the price $p_{\mathrm{A},t}$ to attract more A-type buyers.

We assume that the reservation prices p_{r} of the different buyers are distributed normally with a mean μ_{r} and standard deviation σ_{r}. The pricebot of a seller tries to learn these parameters from the sales data obtained from the A-type buyers. Let $f_t(p_{\mathrm{r}})$ represent the density function for the buyers' reservation prices calculated from the seller's estimate of the parameters $\mu_{\mathrm{r},t}$ and $\sigma_{\mathrm{r},t}$ at the end of interval t. Let $F_t(p_{\mathrm{r}})$ be the corresponding cumulative distribution function obtained by $F_t(p_{\mathrm{r}}) = \int_{\infty}^{-\infty} f_t(p_{\mathrm{r}})\,\mathrm{d}(p_{\mathrm{r}})$. $F_t(p_{\mathrm{A},t})$ then gives the number of buyers whose reservation prices are above $p_{\mathrm{A},t}$. The A-type buyers that purchase from the seller charging $p_{\mathrm{A},t}$ are the buyers whose reservation prices are

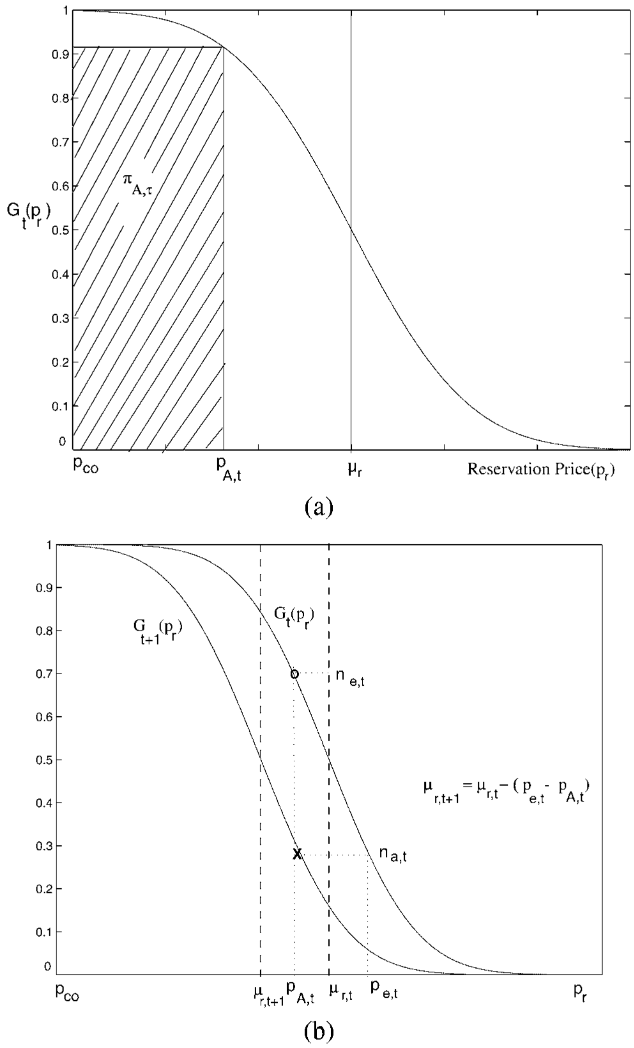


Figure 4. (a) Distribution function $G_t(p_r)$ gives the number of buyers willing to purchase the good at the end of interval t, as a function of the reservation prices of buyers. (b) The pricebot refines its estimate of μ_r using the actual and observed values of the number of customers purchasing at price $p_{A,t}$.

below $p_{A,t}$, and the number of such A-type buyers is given by $1 - F_t(p_r)$. In Figure 4, $G_t(p_r) = 1 - F_t(p_r) = \int_{p_{A,t}}^{\infty} f_t(p_r)\,\mathrm{d}(p_r)$ represents the number of buyers that are still willing to purchase the good when its price is $p_{A,t}$. The profit $\pi_{A,t}$ in Equation (3) is then represented by the area of the rectangle between the current price $p_{A,t}$, the number of buyers $G(p_{A,t})$ purchasing at that price, and the two axes.

Initially, a seller categorizes every buyer as a B-type buyer and places the buyer-id of a first time buyer in the set S_{AB}. When buyers are identified as A-type buyers during an interval t by the consumer segmentation algorithm described in Section 3.1, their buyer-ids are placed in the $S_{A,t}$. The pricebot of the seller then uses the number of buyers in $S_{A,t}$ and the price charged to those buyers $p_{A,t}$ to determine the price $p_{A,t+1}$ for the next interval so that the profit $\pi_{A,t}$ in Equation (3) (i.e., the area of the shaded rectangle in Figure 4) is maximized.

When the seller enters the market, the pricebot is unaware of the values μ_r and σ_r. The pricebot therefore starts with an initial estimate for these values and refines its estimate at the end of every interval as more sales data arrive from the A-type buyers. The pricebot follows the following steps to refine its guess of the initial parameters of the distribution for the buyer reservation prices:

1. Set the initial assumption of the mean of the normal distribution of the buyers' reservation price $\mu_{r,0}$ equal to the market entry price of the seller, i.e., $\mu_{r,0} = p_{A,0} = p_{AB,0}$.
2. At the end of interval t, determine the actual number of A-type buyers that purchased during that interval using the algorithm described in Section 3.1. $n_{a,t} = |S_{p,t} \cap S_{A,t}|$ then gives the number of A-type buyers that purchased from the seller at price $p_{A,t}$ during interval t.
3. Determine the number of A-type buyers that should have purchased at price $p_{A,t}$ using the pricebot's approximation of the distribution $G_t(p_r)$ of the buyers' reservation prices. As shown in Figure 4, $n_{e,t} = G_t(p_{A,t})$ gives the number of A-type buyers that purchase at price $p_{A,t}$ based on the pricebot's current estimates of $\mu_{r,t}$.
4. Refine $\mu_{r,t}$ according to the equation

   $$\mu_{r,t+1} = \mu_{r,t} - (p_{e,t} - p_{A,t})$$

   where $p_{e,t} = G_t^{-1}(n_{a,t})$.
5. Recalculate $G_{t+1}(p_{A,t})$ with the refined value $\mu_{r,t+1}$ and determine $p_{A,t+1}$ as the price that yields

   $$\max\left\{\frac{p_{A,t} - p_{co}}{\sigma_r} \times G_{t+1}(p_{A,t})\right\}.$$

At the end of interval t, the pricebot updates the sets $S_{A,t}$ and $S_{AB,t}$ with the ids of the buyers that purchased during that interval. It then uses steps 2–4 above to refine its estimate of the parameters for the distribution of buyers' reservation prices and to determine $p_{A,t+1}$ for the next interval. As the pricebot adjusts the price being charged to A-type buyers at successive intervals, some A-type buyers cease to purchase as soon as $p_{A,t}$ exceeds their reservation price. Therefore, the value of $n_{A,t}$ keeps changing. Also, at each successive iteration, the pricebot's estimation of $\mu_{r,t}$ is refined. The process converges as $\mu_{r,t}$ approaches the actual value of μ_r.

4. Simulations

The parameters used for our simulations are the seller's cut-off price $p_{co} = 0.1$, $\sigma_r = 0.3$, and the buyer's valuation p_r, is drawn from the normal distribution $[1.0, 0.3]$. For all the simulations, as a crude approximation to an e-commerce survey presented in [Clark, 9], we have considered 250 A-type buyers and 750 B-type buyers.

The price adjustment interval for sellers is taken as $\tau_s = 20$ quote requests from buyers. We have selected the number of quote-requests received from buyers as the unit for measuring time to equalize differences between sellers with different response times. The MO algorithm used by every seller to determine the price for the B-type buyers does not use price and profit information from other sellers. Therefore, the same value of τ_s for every seller does not affect the price calculation in the MO algorithm. A seller that uses longer price-update intervals fails to track the price dynamics in the market as efficiently as a seller that user shorter price-update intervals. Therefore, a seller has to compete with the most number of competitors who track prices as efficiently as the seller itself, when all sellers adjust their prices with the same value of τ_s. The value of $\tau_s = 20$ quotes was determined through repeated simulations with different values τ_s as the optimal value that amortizes the time to run the MO and consumer segmentation algorithms over the time spent in giving out quotes to buyers. Similarly, we have determined by repeated simulations with different values, that a history window size of 5 guarantees the best response of the MO algorithm.

While determining the estimated number of buyers $n_{e,t}$ who are expected to purchase the good at $p_{A,t}$ during interval t, the pricebot uses the function $G_t(p_r)$ given by:

$$G_t(p_r) = 1 - F_t(p_r) = \int_{p_{A,t}}^{\infty} f_t(p_r)\,\mathrm{d}(p_r).$$

The integral of the normal density function $f_t(p_r)$ gives rise to the intractable error function $F(p)$, which is difficult to calculate. We have therefore used approximations for the normal distribution function given by logarithmic tables to calculate the value of $F_t(p_r)$ and $G_t(p_r)$.

Figure 5(a) shows the price variations for three sellers that employ only the MO algorithm. Initially, all three sellers start with a high price. However, they reduce their price in subsequent price adjustments so that each seller can attract the large number of price sensitive B-type buyers and obtain the highest profit. This drives the price of the good down to p_{co} around 5000 quotes and the sellers make little or no profit at this price. In this situation, the sellers can obtain more profits by charging a high price near p_r and attracting only the A-type buyers, instead of operating at p_{co} to attract the B-type buyers. As soon as the sellers discover this, they leave the price war and set a high price. However, at the high price they once again begin to under-cut each other, thereby starting a fresh cycle of the price war.

Figure 5(b) shows the price variations for three sellers using the tiered pricing algorithm. The classification parameter η for distinguishing between A and B-type buyers is set at $\eta = 0.25$. Instead of charging a uniform price for the good from both A and B-type buyers, the tiered pricing algorithm charges a high price from the price-insensitive A-type buyers while it lowers the price for the price-driven B-type buyers. A two-tiered price schedule removes the cycles of price wars seen in Figure 5(a) which were created by the sellers' strategy change from attracting B-type buyers to attracting A-type buyers. The excursions in the high price towards the left of the graph in Figure 5(b) represent the time that the

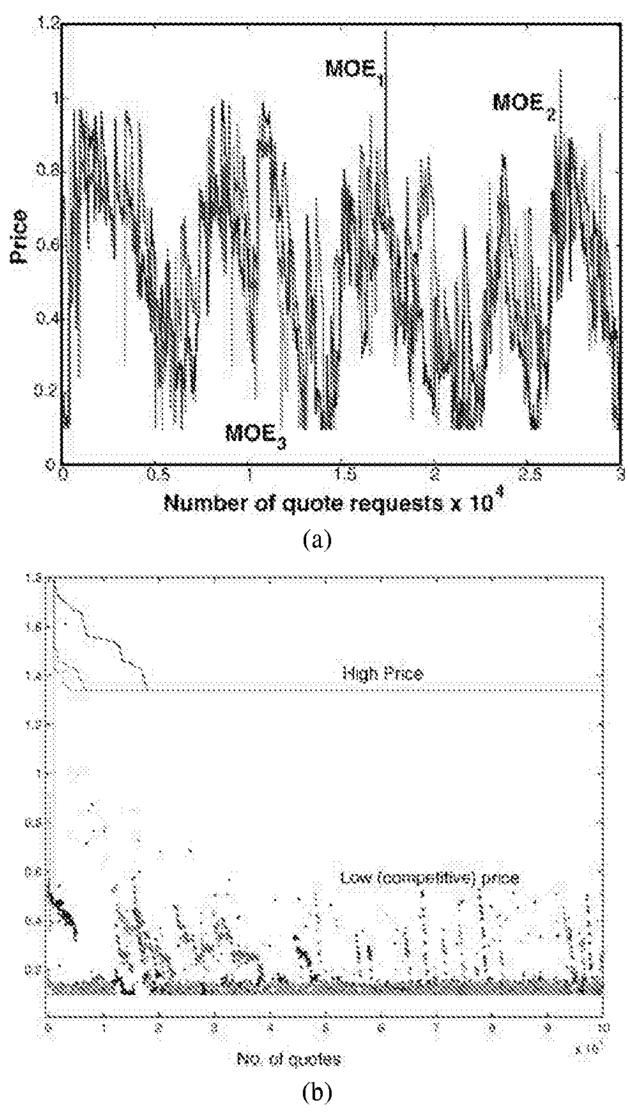
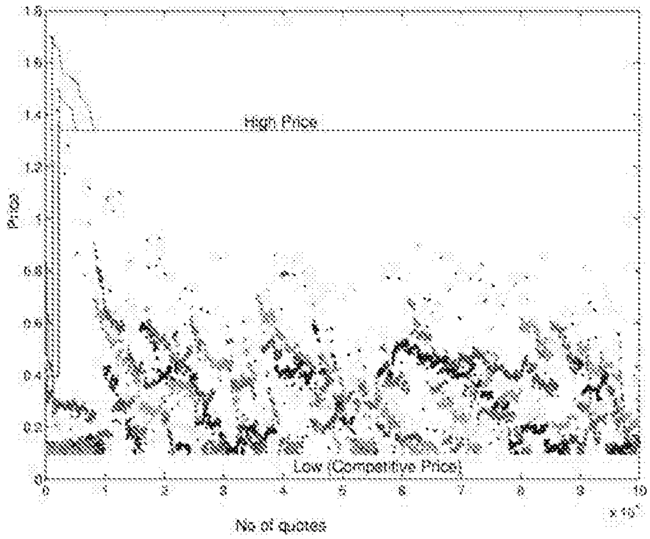
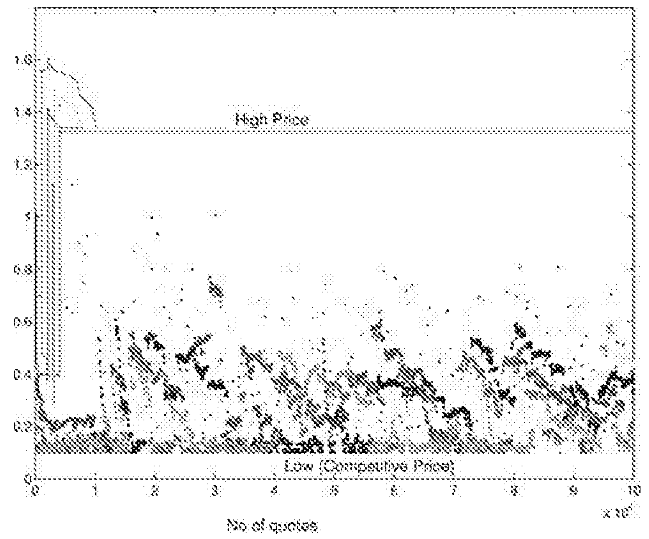


Figure 5. (a) Price vs. no. of quotes profile of a competition between three sellers using the MO algorithm only with uniform pricing for both A and B-type buyers. (b) Price vs. no. of quotes profile of a competition between three sellers with consumer segmentation followed by tiered pricing.

seller spends in guessing the mean of the distribution of cut-off prices for the buyers. The low price that is charged to the buyers illustrates some excursions. This is because, initially all buyers are added to the set S_B. The A-type buyers are insensitive to price and therefore cause the price to increase by small amounts. However, with the passage of time, these A-type buyers get segmented into the set S_A and therefore the excursions in



(a)



(b)

Figure 6. (a) Price vs. no. of quotes profile for 5 sellers using the consumer segmentation algorithm with $\eta = 0.25$. (b) Price vs. no. of quotes profile for 5 sellers using the consumer segmentation algorithm with $\eta = 0.15$.

the low price reduces with time. Figure 6(a) illustrates the price vs. no of quotes profile for five sellers with the same consumer population. As the number of sellers in the market increases, the competition between the sellers to attract the B-type buyers increases. Since the number of buyers in both the simulations remains the same, when the number of sellers increases, increased competition to attract the B-type buyers causes every seller to rely more on A-type buyers for their revenue. Since the price- insensitive A-type buyers cause price fluctuations in the market, the lower price charged by the sellers fluctuates more than when there were fewer sellers. Also, since the sellers rely more on the A-type buyers the time required to determine the mean of the buyers' cut-off prices is less than when there were fewer sellers.

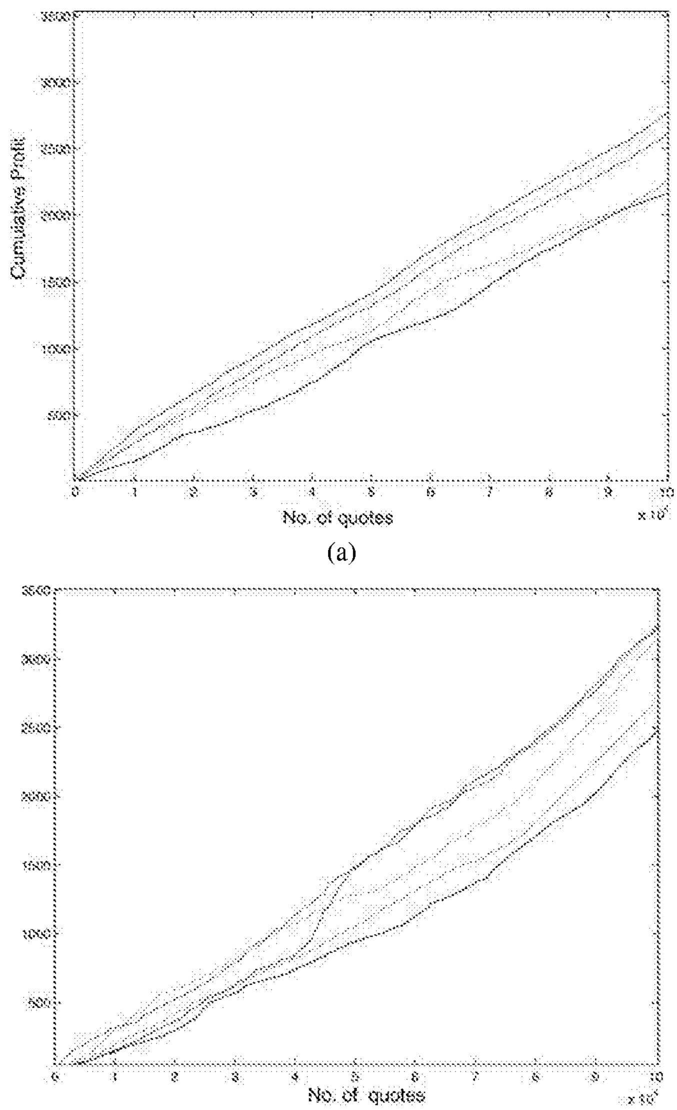
Our next set of simulations illustrate the effects of the value of the classification parameter η on the tiered pricing algorithm. Figure 6(a) shows the price vs. no. of quotes profile of five sellers using the tiered pricing algorithm with $\eta = 0.25$, while Figure 6(b) shows the results with $\eta = 0.15$, all other parameters remaining the same. Since we decreased the value of η from 0.25 to 0.15 the A-type buyers are identified more efficiently. This is illustrated by the reduced excursions in the low price as the A-type buyers are identified quickly and transferred to S_A. However, the lowering of η also causes misclassification of some B-type buyers as A-type. Because B-type buyers constitute 75% of the buyer population, therefore this misclassification results in reduced profits as compared to the profits obtained with $\eta = 0.25$. Therefore, although $\eta = 0.25$ is a modest estimate, it yields better profits than a more accurate estimate of the value of η.

Finally, Figures 7(a) and 7(b) contrast the cumulative profits of five pricebots with and without tiered pricing. In Figure 7(a) every pricebot charges a uniform price to all of the consumers using the MO algorithm only. On the other hand, the pricebots in Figure 7(b) use the tiered pricing algorithm and obtain 15% more cumulative profit than the corresponding pricebots in Figure 7(a). Thus, the tiered pricing for different consumer segments is successful in increasing the profits to the sellers.

5. Conclusion

In this paper, we have presented a consumer segmentation algorithm that separates buyers into price-sensitive and price-insensitive categories depending upon the purchase rates of the buyers. We then use the model optimizer algorithm to determine the price to be charged by the seller to the price-sensitive buyers. Also, the pricebot of the seller tries to learn the distribution of the buyers' reservation prices and identify the price that should be charged to the price-insensitive buyers so that the profits to the seller are maximized. Our simulations indicate that by using a tiered pricing strategy, a seller can increase its profits by 15–20% than what it can obtain without segmenting the consumers based on their purchase preferences.

Consumer segmentation has already been tested by some online sellers like Amazon [2]. Although implemented successfully, tiered pricing was criticized for being unfair to price-insensitive buyers. However, we believe that, if tiered pricing is combined with promotions on other attributes, price-insensitive customers will be willing to pay a higher price for added privileges on other parameters. Some online stores have already started to profile its customers' preferences so that they can offer premium service, albeit at an elevated price, when the customers return. In the future, we intend to extend the algorithm described in



(a)

(b)

Figure 7. (a) Cumulative profit of five pricebots using the MO algorithm only with uniform pricing for both A and B-type buyers. (b) Cumulative profit of five pricebots with consumer segmentation followed by tiered pricing.

this paper to include multi-attribute based consumer segmentation and customer-profile based pricing.

In our model, price-sensitive buyers always purchase from the seller offering the lowest price for an item in the market. However, the price sensitive nature of buyers can be affected by certain factors in the market. Here, we have assumed that buyers do not possess any memory of their purchases. Nor do they share any information about their purchase with other buyers in the market. However, in the real world, consumers are influenced by their previous experiences while making a purchase decision. In such a scenario, price-sensitive buyers can exhibit loyalty to certain sellers and continue to purchase from them despite of their price-sensitive nature. Although online buyers do not actively exchange information with one another, merchant referral sites such as Bizrate [5] and Reseller Ratings [29] provide consumers with ratings for merchants on different attributes including promotional offers, promptness of delivery and customer service. Buyers that are informed about a merchant's performance display affinity to purchasing from sellers with a good reputation disregarding their inherent price-sensitive nature. The influence of additional factors such as purchase history and recommendations about sellers on the purchase decision of buyers is a direction for our future research.

The sellers in our model do not employ any information about their competitors. In the real world, the price information from competitors is often used by a seller to determine the price that it will charge. However, frequent requests for prices from competitors slows down the operation of the seller. Besides, malicious sellers can obstruct competitors' sales by harassing them frequently with price requests. A seller can discover competitors' price relatively infrequently and employ machine learning techniques to forecast patterns in the competitors' pricing strategy. The seller can then integrate the competitors' pricing strategy to determine the price that it will charge. Price information from competitors is likely to increase the competition between sellers in the market. In the future, we intend to combine machine learning techniques in a seller's pricing strategy so that it can determine the price that it should charge more efficiently.

In this paper, we have discussed a rather naive model of buyers and sellers to determine the effects of dynamic tiered pricing strategies used by sellers. Our experiments exhibit that consumer segmentation based on buyer preferences followed by tiered pricing increases the profits to sellers. In the future, we intend to extend our work along the directions mentioned above to make buyers and sellers more responsive to the dynamics in an online market.

References

[1] Alta Vista Inc. URL: `http://www.altavista.com`.

[2] Amazon Inc. URL: `http://www.amazon.com`.

[3] Batson, A. (1998). "Win-Win Interactions Between the Public and Private Sectors." *Natural Medicine, Vaccine Supplement* 4(5) (supplement), 487–491.

[4] Ben Schafer, J., J. Konstan, and J. Riedl. (2000). "Electronic Commerce Recommender Applications." *Journal of Data Mining and Knowledge Discovery* 5(1/2), 115–152.

[5] Bizrate Inc. URL: `http://www.bizrate.com`.

[6] Breese, J., D. Heckerman, and C. Kadie. (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering." In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, pp. 43–52.

[7] Brooks, C., S. Fay, R. Das, J.K. Mackie-Mason, J. Kephart, and E.H. Durfee. (1999). "Automated Search Strategies in an Electronic Goods Market: Learning and Complex Price Scheduling." In *Proceedings of the 1st ACM Conference on Electronic Commerce*.

[8] Caglayan, A., M. Snorrason, J. Jacoby, J. Mazzu, and R. Kumar. (1997). "Learn Sesame: A Learning Agent Engine." *Applied Artificial Intelligence* 11, 393–412.
[9] Clark, D. (2000). "Shopbots Become Agents for Business Change." *IEEE Computer* 33, 18–21.
[10] Cohen, S. and V. Ramaswamy. (1998). "Latent Segmentation Models." *Marketing Research* 10(2), 14–21.
[11] Dasgupta, P. and R. Das. (2000). "Dynamic Service Pricing Using Brokers in a Multi-Agent Economy." In *Proceedings of the Fifth International Conference on Multi-Agent Systems*, Boston, MA, pp. 375–376.
[12] Dasgupta, P. and R. Das. (2000). "Dynamic Pricing with Limited Information in a Multiagent Economy," In *Proceedings of the Fifth International Conference on Cooperative Information Systems*. Eilat, Israel, pp. 299–310.
[13] Google Inc., URL: `http://www.google.com`.
[14] Greenwald, A., J.O. Kephart, and G.J. Tesauro. (1999). "Strategic Pricebot Dynamics." In *Proceedings of 1st ACM Conference on E-Commerce*.
[15] Herlocker, J., J. Konstan, A. Borchers, and J. Riedl. (1999). "An Algorithmic Framework for Performing Collaborative Filtering." In M. Hearts, F. Gey, and R. Tong (eds.), *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, pp. 230–237.
[16] Kay, J. (1995). "The *um* Toolkit for Reusable, Long Term User Models." *User Modeling and User-Adapted Interaction* 4(3), 149–196.
[17] Kephart, J. and A.R. Greenwald. (1999). "Shopbot Economics." In *Proceedings of the Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*.
[18] Kephart, J., J. Hanson, and A. Greenwald. (2000). "Dynamic Pricing by Software Agents." *Computer Networks* 32(6), 731–752.
[19] Kramer, M. (2001). "ATG's Dynamo Scenario Server: Scenario-Based E-Commerce, CRM, and Merchandising." White paper. URL: `http://www.atg.com/repositories/ContentCatalogRepository_en/white_papers/PSGReport.pdf`.
[20] Manna Inc. URL: `http://www.mannainc.com/products.html`.
[21] Moai. URL: `http://www.moai.com`.
[22] My Simon. URL: `http://www.mysimon.com`.
[23] Nelder, J. and R. Mead. (1965). "A Simplex Method for Function Minimization." *Computer Journal* 7, 308–313.
[24] Net Perceptions. URL: `http://www.netperceptions.com`.
[25] Orwant, J. (1995). "Heterogenous Learning in the Doppelganger User Modeling System." *User Modeling and User-Adapted Interaction* 4(2), 107–130.
[26] Paiva, A. and J. Self. (1995). "TAGUS: A User and Learner Modeling System." *User Modeling and User-Adapted Interaction* 4(3), 197–226.
[27] Press, W., S. Teukolsky, W. Vetterling, and T. Flannery. (1988). *Numerical Recipes in C*. Cambridge, MA: Cambridge University Press.
[28] Pricing Central. URL: `http//www.pricingcentral.com`.
[29] Reseller Ratings Inc. URL: `http://www.resellerratings.com`.
[30] Site Sell Inc. URL: `http://www.sitesell.com`.
[31] Talus Solutions. URL: `http://www.talussolutions.com`.
[32] Tesauro, G. and J.O. Kephart. (1999). "Pricing in Agent Economies Using Multi-Agent Q-Learning." In *Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*.
[33] Trivedi, M. (1999). "Using Variety Seeking Based Segmentation Models." *Journal of the Academy of Marketing Science* 27(1), 37–49.