Multi-attribute Regret-based Dynamic Pricing

Janyl Jumadinova*, Prithviraj Dasgupta

Computer Science Department
University of Nebraska, Omaha, NE 68182, USA
E-mail: {jjumadinova, pdasgupta}@mail.unomaha.edu
*:Primary author, student

Abstract. In this paper, we consider the problem of dynamic pricing by a set of competing sellers in an information economy where buyers differentiate products along multiple attributes, and buyer preferences can change temporally. Previous research in this area has either focused on dynamic pricing along a limited number of (e.g. binary) attributes, or, assumes that each seller has access to private information such as preference distribution of buyers, and profit/price information of other sellers. However, in real information markets, private information about buyers and sellers cannot be assumed to be available *a priori*. Moreover, due to the competition between sellers, each seller faces a tradeoff between accuracy and rapidity of the pricing mechanism. In this paper, we describe a multi-attribute dynamic pricing algorithm based on minimax regret that can be used by a seller's agent called a *pricebot*, to maximize the seller's utility. Our simulation results show that the minimax regret based dynamic pricing algorithm performs significantly better than other algorithms for rapidly and dynamically tracking consumer attributes without using any private information from either buyers or sellers.

Keywords: Dynamic pricing, pricebots, minimax regret.

1 Introduction

With the increasing automation of e-commerce applications, intelligent agents are becoming an essential part of various business transactions. Over the past decade, several services such as automated comparison shopping tools including MySimon[2] and PriceGrabber[3], and seller ratings Websites such as Bizrate [1] have enabled online buyers make rapid and informed decisions before purchasing products over the Internet. As the number of buyers who rely on these services increases, it is becoming advantageous for online sellers to use automated pricing-setting techniques in an attempt to maximize profits. Intelligent agents called *pricebots* [14] provide a suitable paradigm for online sellers to rapidly update the price of a product in response to changes in market parameters such as buyer preferences in an online economy.

Consumers who purchase products online are frequently willing to pay an elevated price for enhanced values on particular product attributes such as delivery time, seller reputation, and after-sales service[1]. Different consumers have also been reported to prefer different product attributes and these preferences vary over time depending on exogenous factors such as sales promotion, aggressive advertising and even time of the year [16]. Therefore, it is important for an online seller to differentiate a product it sells along multiple attributes, and, determine a potential buyer's purchase preferences over the different product attributes, so that it can tailor its offer to meet the buyer's requirements, and, improve its profits. Online markets are also characterized by multiple sellers for the same product. To remain competitive in such a market, a seller has to offer a price that is more attractive than its competitors prices to potential buyers. To achieve this, each seller has to use a dynamic pricing algorithm that calculates a profit maximizing price for the seller. Previous research on dynamic pricing algorithms requires each seller to possess *a priori* information about the market parameters such as buyers' reservation prices, the number of buyers, preferences of buyers over different product attributes and prices and profits of other competing sellers in the market. However, in real-life economies accurate knowledge of such market parameters cannot be assumed to be available with sellers. In this paper, we make two contributions to the problem of dynamic pricing in a market where buyers differentiate products along multiple attributes. First, we describe a preference elicitation algorithm based on minimax regret that can be used by a seller's pricebot to determine the distribution of buyer preferences along different product attributes. Then, we describe a minimax regret based algorithm that enables a seller to dynamically update the posted price of a product to improve its profit. Both these algorithms do not require any *a priori* knowledge about market parameters such as buyer's preferences over product attributes, buyers' reservation prices and other competing sellers' prices and profits. These algorithms only require a seller's private history including its posted prices, profits and purchase decisions from different buyers for their calculations. Our simulation results show that the minimax regret-based attribute prediction algorithm is able to predict the preferred attributes of different buyers with more than 90% accuracy in most cases, even when the buyers' preferences over different attributes change dynamically over time. When used with the attribute prediction algorithm in a competitive market, a seller using the minimax regret based dynamic pricing algorithm is able to obtain $9-13\%$ more profits than competing sellers using other dynamic pricing strategies.

2 Dynamic Pricing over Multiple Product Attributes

Current real-life internet economies involve complex interactions between several buyers, sellers and possibly brokers that facilitate trading. We have made certain simplifying assumptions of an online economy to simplify analysis while retaining the essential features of the market. Our online market model is based on the shopbot economy model of Kephart and Greenwald [14]. We consider an economy that consists of a set of S sellers who compete to provide a set of B buyers with a single homogeneous product, where $\mid B \mid \gg \mid S \mid$. Each seller behaves as a profit maximizer and has a sufficient supply of the product to last the lifetime

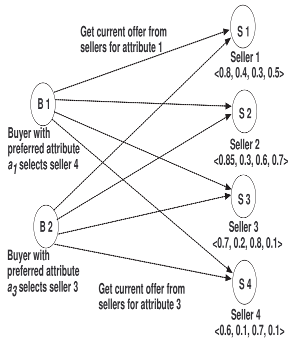

Fig. 1. A hypothetical market showing two buyers with preferred attributes as a_1 and a_3 respectively making a quote request to four sellers and selecting the seller that offers the best price for the product on their respective attributes. The tuple $< p_{a_1}, p_{a_2}, p_{a_3}, p_{a_4} >$ below each seller denotes the normalized price offered by each seller on the different product attributes.

of the buyers. Buyers come back to the market repeatedly to re-purchase the product. A product is differentiated by buyers and sellers on multiple attributes such as offered price, delivery time, product quality, seller reputation, previous experience with seller, etc. Here, we assume that each product has a set of A different attributes, and, the buyers and sellers in the market are aware of this set of product attributes.

Every buyer in the market selects one of the product's attribute as its preferred attribute and is willing to pay a slightly elevated price to purchase the product along its preferred attribute. In online markets, buyers also change their preferred attribute for a product dynamically [16]. For example, when buyers purchase products under time constraints, they prefer the 'delivery- time' attribute of the product. On the other hand, when buyers do not consider time as a critical factor, product quality or seller reputation are possible attributes that determine their purchase decision. The choice of the preferred product attribute of a buyer can also be affected by exogenous factors such as time of the year, previous purchase experiences, etc.[18]. To model this, we assume that every buyer in our market has one of the product's attributes as its preferred attribute and the preferred attribute of a buyer can vary with time. Sellers are unaware of buyers' preferred attributes and the temporal variation of the preferred attributes for each buyer. A seller offers a slightly different price for the product along each product attribute $a_i \in A$. To improve its profits in a market with multiple sellers offering the same product, each seller has to dynamically adjust its posted prices along the different product attributes so that it can continue to offer a competitive price and attract buyers. When a buyer requests a price quote, the buyer's preferred attribute is not known to a seller. Therefore, the objective of a profit maximizing seller is to determine a buyer's preferred attribute in response to the buyer's quote request. The seller can then calculate a competitive price of the product along the buyer's preferred attribute and make an attractive offer to the buyer.

An illustration of the operation of our market is shown in Figure 1. A seller j enters the market with an initial price $p^0_{a_i,j}$ for a unit of the product under attribute a_i. Each seller has a unit production cost p_{co} below which it is not willing to sell the product. A buyer wishing to purchase a product first requests a price quote from the sellers. We assume that buyers use comparison shopping services [3] to discover sellers and are aware of all the sellers in the market. Since we analyze pricing algorithms for sellers, seller discovery is not treated as a major issue in the model. Each seller j receiving a buyer's quote request responds with a price vector $\bar{P}^t_j = < p^t_{a_i,j} >$, where $p^t_{a_i,j}$ represents the price charged by seller j during interval t along product attribute a_i. This price is updated by the seller's pricebot at intervals τ_j using a dynamic pricing algorithm. Different sellers update their product prices asynchronously and each seller uses its own pricing strategy. When a buyer that had made quote requests receives the offers from different sellers, it compares the offers made by the different sellers. Each buyer b has a reservation price for the product p_{r,b,a_i} along attribute a_i, above which it is not willing to buy the product. The utility of the product to a buyer b from seller j along attribute a_i is given by $U_{b,a_i,j} = p_{r,b,a_i} - p^t_{a_i,j}$, where $p^t_{a_i,j}$ is the posted price of the product offered by seller j along attribute a_i during interval t. The purchase decision is made by buyer b by comparing the utilities from the different sellers along the preferred attribute a_i of the product, and, selecting seller S_k given by $S_k = \arg_S \max U_{b,a_i,S}$. In case, more than one seller offers the same lowest price along the buyer's preferred attribute, one of the sellers offering the highest utility is chosen randomly by the buyer. Buyer b then pays seller S_k the posted price of the product and the seller delivers the product. Payment and product delivery are not discussed any further here as we concentrate on seller's preferred attribute prediction and pricing strategies. In the next two sections, we present the minimax regret-based dynamic pricing algorithm that is used by sellers to estimate buyers' preferences and dynamically update the posted prices over the different product attributes.

3 Minimax Regret-based Algorithms

The parameters used in our market model are given below:

A	Set of product attributes.
B	Set of buyers.
S	Set of sellers.
p_{r,b,a_i}	Buyer b's reservation price along attribute a_i.
p_{co}	Unit production cost for a seller(assumed to be uniform over all attributes)
τ_j	Price update interval for seller j.
$p^t_{a_i,j}$	Posted price of the product by seller j along attribute a_i during interval t.
$\pi^t_{a_i,j}$	Profit obtained by seller j along attribute a_i during interval t.
u^t_{b,a_i}	Upper bound on the buyer's purchase value for attribute a_i.
l^t_{b,a_i}	Lower bound on the buyer's purchase value for attribute a_i.

3.1 Minimax Regret-based Attribute Prediction

To remain competitive in a market where buyers can change their preferred attribute, each seller should attempt to accurately determine the current preferred attribute of the buyers that request price quotes from it, so that the seller can charge a profit maximizing price along the buyers' preferred attribute. However, dynamically determining the buyers' preferred attribute without any knowledge of the buyers' demand and attribute variance function is a challenging task facing the seller. In this paper, we describe a preference elicitation based technique that can be used by sellers to predict the buyers' preferred attribute.

Most of the previous literature on user preference elicitation uses considerable information extracted from users to determine the user's preferences over different choices. However, in our model there are three challenges in collating consumer data for analysis by sellers for preference elicitation: (1) A buyer's preferred product attribute changes over time and a seller needs to continuously update the buyer's preferences over the different product attributes to be able to determine the current preferred attribute of the buyer. (2) A seller and a buyer interact for a limited duration and the only information that the seller is able to get from a buyer is the purchase decision (yes or no) of the buyer from that seller. (3) Because a seller is not aware of the prices offered by other sellers in the market, when a buyer does not purchase a product from the seller after receiving the seller's offer, the seller does not have any mechanism for inferring whether the negative purchase decision resulted from incorrect calculation of buyer's preferred product attribute, or, whether another seller offered a more attractive price to the buyer along the buyer's preferred attribute.

To address these issues, we first make the observation that elicitation of the full buyer preferences captured by the utility function of the buyer might be unnecessary in determining the buyer preferred product attribute. A reasonable estimate of the buyer preferences can be obtained by a seller from the purchase decision made by a buyer after the buyer requests a quote from the seller. To elicit the buyer preferences from the purchase decision information of a seller, each seller in our model uses the minimax regret technique of preference elicitation described by Boutilier *et al.* in [6]. To enable the preference elicitation mechanism, each seller assumes that there is a set of bounds on every buyer's expected purchase value of the product. These bounds keep track of the price levels at which a buyer will purchase the product and can be used as an indicator for both the buyer's valuation(reservation price) of the product as well as the prices charged by the competing sellers at which the buyer has recently purchased the product. The minimax decision criterion suggests that the seller makes a decision that gives the minimum max-regret, where max-regret is the largest value by which the seller could regret making that decision. It is therefore a decision the seller would regret the least and minimizes the worst-case loss the seller would encounter after making that decision.

In our model, sellers have to make a decision at the end of each interval about which attribute to predict for each buyer. To realize this, each seller keeps an upper bound u_{b,a_i} and a lower bound l_{b,a_i} for every buyer on their expected purchase values for each attribute of the product. To enable the elicitation of buyers' preferences, we consider the buyer-seller interaction as a querying process. Essentially sellers are sending a query to a buyer every time they respond to that buyer's quote request with the product's posted prices.

```
function minimax_attrib_predict returns
    input: int $\tau_j$;            // time interval length (in quotes)
    variables: int $qr$;           // number of quote requests received by seller
            int $t$;               // time interval
            double $p^t_{a_i,j}$;     // seller j's price along attribute $a_i$
            set $B^t_j$;           // set of buyers that have accessed $j$ during interval t
            set $B[a_i]$;          // index set of buyers under attribute $a_i$
            $t \leftarrow 0$; $B^t_j \leftarrow \emptyset$;
    while(seller remains in market)
        for every $a_i \in$ A
            $p^t_{a_i,j} \leftarrow p \mid p \in U[p_{co}, 1]$
        while ($qr < \tau_j$)
            if some buyer $b \in B$ requests quote from seller
                $qr \leftarrow qr + 1$;
                $B^t_j \leftarrow B^t_j \bigcup b$;
            if some buyer $b' \in B^t_j$ purchases product with preferred attribute $a_i$
                $B^t_{j,a_i,pos} \leftarrow B^t_{j,a_i,pos} \bigcup b'$

        for every buyer $b \in B^t_j$
            for every $a_i \in A$
                if(b $\in B^t_{j,a_i,pos}$)
                    $l^t_{b,a_i} \leftarrow l^t_{b,a_i} + \epsilon$;
                else
                    $u^t_{b,a_i} \leftarrow u^t_{b,a_i} - \epsilon$;
                    $B^t_{j,a_i,neg} \leftarrow B^t_{j,a_i,neg} \bigcup b$;
        // condition $u_{b,a_i}$ and $l_{b,a_i}$ values
        for every $a_i \in$ A
            $u^t_{b,a_i} \leftarrow \sum_{k=0}^{h} \lambda_k u^{t-k}_{b,a_i} | \sum \lambda_k = 1, \lambda_{k-1} > \lambda_k$;
            $l^t_{b,a_i} \leftarrow \sum_{k=0}^{h} \lambda_k l^{t-k}_{b,a_i} | \sum \lambda_k = 1, \lambda_{k-1} > \lambda_k$;
        for every buyer $b \in B^t_j$
            for every $a_i \in A$
                $R(a_i, a_{-i}) \leftarrow u^t_{b,a_{-i}} - l^t_{b,a_i}$;
                $R(a_i, a_i) \leftarrow 0$;
                for every $a_i \in A$
                    $MR_{a_i} \leftarrow max_{a_{-i} \in A}\ R(a_i, a_{-i})$;
                $a_l \leftarrow arg_{a_i} min\ MR_{a_i}$;
            Place buyer $b$ in $B[a_l]$;
            Remove buyer $b$ from any other $B[a_{-l}]$;
        $t \leftarrow t + 1$;
```

Fig. 2. Algorithm used by the sellers to predict buyers preferred attributes at the end of each time interval.

We can consider this query as a bound query of type: "Is your valuation of the product greater than or equal to my offered price?" Given a "yes" response, the seller modifies that buyer's lower bound and given a "no" response, the seller adjusts buyer's upper bound, thus tightening the bounds of the buyer's purchase valuation. The algorithm describing the attribute prediction process is presented in Figure 2. Initially, seller j sets its price along each attribute randomly (lines 1-2). Seller j keeps track of the number of quotes it receives and records whether a purchase was made or not by the buyers (lines 3-9). When the number of quote requests received by seller j reaches τ_j (time interval for price update measured in number of buyer quote requests), seller j updates its bounds on the purchase valuation of every buyer that has purchased from it over the current time interval. A positive purchase decision by a buyer raises the lower bound of the purchase valuation while a negative purchase decision lowers the upper bound of the purchase valuation (lines 10-16). Seller j then weighs both of these bounds over the historical values of the previous h bounds used by it, with higher weights given to more recent bounds, to prevent wide fluctuations in these values(lines 17-19). To calculate the minimax regret, seller j first calculates the pairwise regret $R(a_i, a_{-i})$ of attribute $a_i \in A$ with respect to other attributes $a_{-i} \in A$. This value corresponds to the regret the seller feels for predicting attribute a_i instead of any other attribute a_{-i}. Seller j then selects the attribute a_l corresponding to the minimum of the maximum regrets from these pairwise regret values as the preferred attribute for buyer b. This calculation is repeated for every buyer b that purchases from the seller in the current time interval (line 22-26).

3.2 Regret-based Dynamic Pricing

At the end of each time interval, seller j predicts the buyer's preferred attribute and then updates its price along each attribute a_i. A seller's objective is to calculate a profit maximizing price along each product attribute while considering the number of buyers that were determined to have that attribute as its preferred attribute during that time interval using the attribute prediction technique in Section 3.1. The algorithm for achieving this dynamic pricing is described in Figure 3. Seller j first calculates the average bounds on the purchase valuations across all buyers for each product attribute (lines 1-2). It then calculates a historical weighted average price $\overline{p}^t_{a_i,j}$ using prices in h previous intervals (line 3). The seller then calculates the normalized number of buyers, n_{a_i}, with preferred attribute a_i, using the number of buyers under each product attribute determined by the *minimax_attrib_predict* function (line 27, Figure 2). This value is then used to determine the regret-based price ${p'}^t_{a_i,j}$ by seller j (lines 4-5).

Since the goal of the seller is to maximize its profit, the seller keeps track of its profit direction changes and adjusts its prices so that the profits are increasing. The seller observes the direction of price movement predicted by the average regret-based price and the historical average of prices. If the direction of this price movement is the same as the direction of the profit change in the last interval, the seller sets the new posted price during interval $(t + 1)$ along attribute a_i as the weighted average of regret-based price and historical average price with the larger weight given to regret-based price. Regret-based price accounts for the buyers' predicted preference distribution, whereas historical average price is used to account for some "noise" in the market, which can make the buyers' predicted preference distribution less accurate. Using past price trends, sellers can eliminate sudden changes in the prices that are caused by the inaccurate prediction of the buyers' preference distribution. On the other hand, if the direction of predicted price movement and the direction of profit change are opposite to each other, the seller will still want to update its prices based on the profit changes, since that will yield it more profit. The opposite direction of the predicted price movement to the profit change can happen as a result of some error in the attribute prediction algorithm or some noise in the market. In this case, the seller sets the posted price during interval $(t+1)$ as the price during the last interval t plus a small amount ϵ in the direction of the profit change (lines 6-9).

function minimax_regret_pricing return double[ ] $< p^t_{a_i,j} >$
for every $a_i \in A$
1. $\overline{u}^t_{b,a_i} \leftarrow \sum_{b \in B[a_i]} u^t_{b,a_i} / |B[a_i]|$;
2. $\overline{l}^t_{b,a_i} \leftarrow \sum_{b \in B[a_i]} l^t_{b,a_i} / |B[a_i]|$;
3. $\overline{p}^t_{a_i,j} \leftarrow \sum_{k=0}^{h} \lambda_k p^{t-k}_{a_i,j} \mid \sum \lambda_k = 1, \lambda_{k-1} > \lambda_k$; // historical average
4. $n_{a_i} \leftarrow \frac{|B[a_i]|}{|B|}$;
5. ${p'}^t_{a_i,j} \leftarrow n_{a_i} \overline{u}^t_{b,a_i} + (1 - n_{a_i}) \overline{l}^t_{b,a_i}$; // average regret-based price
6. if $(sign({p'}^t_{a_i,j} - \overline{p}^t_{a_i,j}) * sign(\pi^t_{a_i,j} - \pi^{t-1}_{a_i,j}) = 1)$
7. $\quad p^{t+1}_{a_i,j} \leftarrow \lambda_1 {p'}^t_{a_i,j} + (1 - \lambda_1) \overline{p}^t_{a_i,j}$;
8. else
9. $\quad p^{t+1}_{a_i,j} \leftarrow p^t_{a_i,j} + sign(\pi^t_{a_i,j} - \pi^{t-1}_{a_i,j}) * \epsilon$;
return $< p^{t+1}_{a_i,j} >$

Fig. 3. Minimax regret-based dynamic pricing algorithm used by sellers to update prices at the end of each interval

4 Simulation Results

We have tested our minimax regret based attribute prediction and dynamic pricing algorithm within a simulated market economy. All simulation results have been averaged over 10 simulation runs. Following is a list of parameters and their values we have used in our simulations:

4.1 Comparison Strategies

To quantify the performance of our minimax regret based algorithm with other algorithms for dynamic pricing, we have compared the minimax regret based algorithm with the following strategies: **(1) Fixed Pricing.** In fixed pricing, a

Parameter	Value
Number of buyers	500 or 1000
Number of sellers	3 or 5
Number of product attributes	5
Rate at which buyers send quote requests to sellers	5000 ms
Unit production cost for seller	0.1
Entry price of sellers in market	$U[p_{co}, 1.0]$
Number of past intervals, h	10
Interval for price updates for seller	40 quote requests from buyers [1]
Weight of average regret based price (λ_1 in line 7, Figure 3)	0.8

Table 1. Parameters used for the simulation experiments

seller does not change the posted price of a product. **(2) Derivative Follower Pricing Strategy.** In the derivative follower (DF) strategy, a seller determines the price for the next interval based on the profits obtained from the pricing in the current interval. The price update equation along attribute a_i used by a seller j using the derivative follower strategy is given by: $p^{t+1}_{a_i,j} = p^t_{a_i,j} + sign(\pi^t_{a_i,j} - \pi^{t-1}_{a_i,j}) * \epsilon$, where $\epsilon \in U[0.1, 0.2]$. **(3) Goal Directed Strategy.** In the goal-directed pricing strategy described by DiMicco *et al.*[10], a seller calculates the average number of a units of the product it should sell per interval that enables it to clear the inventory by its last interval in the market. It then observes the number of units it is able to sell during the current interval. If the actual number of units sold is above(below) the expected clearance value, the seller responds by raising(lowering) the price of the product for the next interval.

Variation of Buyer Attribute Preferences. Buyers have a set of discrete probability distributions P_n according to which they vary their preferred attributes. When a buyer enters the market, it randomly selects one of the probability distributions, $p_n \in P_n$. Each probability distribution consists of $\mid A \mid$ probabilities, $p_n =< p_{a_i} >\mid i = 1, ..., \mid A \mid$. Each p_{a_i} corresponds to a buyer's probability of selecting that attribute as its preferred attribute. To model the temporal variation in preferred attributes, each buyer changes its selected probability distribution from p_n to $p_{n'} \in p_{-n}$ at random times.

Attribute Prediction Algorithms. For comparing the minimax regret based attribute prediction algorithm, we have used a collaborative filtering based attributed prediction technique used in dynamic pricing. Collaborative filtering algorithms [17] are used extensively in recommender systems for recommending products to a user by matching the user's preferences along different product attributes with the preferences of other users collected over time. [5, 9] have employed collaborative filtering techniques to determine consumer attributes for the dynamic pricing problem. In these collaborative filtering mechanisms, each seller predicts the buyers' preferred attributes based on the purchase history of buyers with that seller. To achieve this, each seller associates each product

attribute with a cluster of buyers. For every buyer that has purchased from the seller, the seller calculates a set of probabilities $W_t = w_i^t$ for placing the buyer under cluster(attribute) $a_i \in A$ during interval t. Each seller updates these probabilities at the end of every interval based on the purchase decision of the buyer during the interval and the historical values of the probabilities, as outlined below: (1) Update w values: $w_i^{t+1} = (w_i^t)^{\frac{p_i^t}{p} - \frac{n_i^t}{n}}$, where p_i^t is number of purchases during interval t along cluster i and n_i^t is the number of no-purchases, p is the total number of purchases along attribute i and n is the total number of no-purchases.
Find w_i^{t+1} values for each attribute, $i = 1.. \mid A \mid$ and put them into the W_{t+1} vector (2) Find the cosine similarities of W_{t+1} with all previous cluster probability vectors that are in the history table according to the following equation: $sim = \frac{W_{t+1}.W_j}{||W_{t+1}||*||W_j||}$, where j represents elements in the history table. (3) Select the probability vector W_{t-j} that is most similar to W_{t+1} and calculate of weighted sum of W_{t-j} and its h successive probability vectors: $W'_{t+1} = \Sigma_{k=0}^{h} \lambda_k W_{t-j+k}$ (4) The probability values in W'_{t+1} are then used to assign buyers into clusters during next interval $t+1$.

4.2 Experimental Results

Attribute Prediction. In our first set of simulations, we compare the attribute prediction accuracy, independent of price setting, using the collaborative filtering and the minimax regret-based techniques. Figure 4(a) shows the attribute prediction comparison along attribute $a_2 \in A$ in a market with 3 sellers and 500 buyers, where two of the sellers use the minimax regret based attribute prediction while the remaining seller uses the collaborative filtering technique for attribute prediction. We observe that sellers using the minimax regret-based attribute prediction technique are able to predict the number of buyers with preferred attribute a_2 within $0 - 15\%$ accuracy of the actual number of buyers under the attribute. Although the collaborative filtering based technique performs comparably, it shows intermittent excursions in the attribute prediction resulting in the preferred attribute being incorrectly predicted for as many as 80% of the buyers in the market. The relative inaccurate predictions of the collaborative filtering algorithm can be attributed to the fact that collaborative filtering performance is highly dependent on other buyers purchase decisions and might introduce biased effects. This causes the overall most preferred attributes to be recommended more often and prevents the seller from adjusting to changes in buyer preferences. In Figure 4(b), we show the comparison results for the two attribute prediction techniques in a market with 5 sellers and 1000 buyers, where three of the sellers use the minimax regret based attribute prediction while the remaining two sellers use the collaborative filtering technique for attribute prediction. Once again, we observe that the minimax regret based technique is able to predict the attribute of the buyers accurately most of the time and has a maximum error of only 10% during the entire simulation. On the other hand, the collaborative filtering based technique performs considerably poorly with errors ranging from

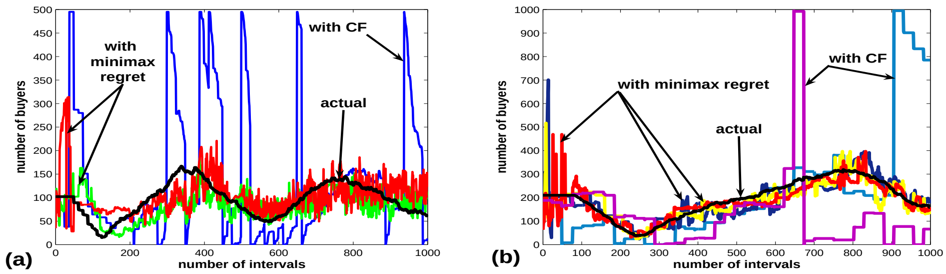

Fig. 4. Attribute prediction using minimax regret based attribute prediction and collaborative filtering techniques. (a) In a market with 3 sellers and 500 buyers and (b) in a market with 5 sellers and 1000 buyers.

$20-30\%$ through most of the simulation. The reason for the poor performance of the collaborative filtering algorithm can be attributed to the probability values varying significantly with a big change in the number of purchases which results in most buyers being classified under one attribute. This contributes to sellers using collaborative filtering predicting one attribute for most of the buyers during some intervals and resulting in large inaccurate oscillations in the number of predicted buyers The average prediction accuracy for our simulations in Figure 4 was 89% for minimax regret-based attribute prediction and 69% for attribute prediction using the collaborative filtering approach.

Dynamic Pricing. In our next set of simulations we compared the performance of different pricing strategies used by different sellers in our simulated market economy. For the first set of experiments, we used a market with 500 buyers and 3 sellers. Two of the sellers used the minimax regret based dynamic pricing strategy while the remaining seller used the strategy being compared. Figure 5(top) shows the profit comparisons and price competition of two sellers using the minimax regret-based dynamic pricing and one seller who sets a fixed price. In our simulations, fixed-price sellers set the price for each attribute randomly when they enter the market. Figure 5(top) illustrates that even when the fixed price is initially below the prices set randomly by the minimax regret-based dynamic pricing sellers, sellers using minimax regret-based pricing strategies adjust their prices as they compete with each other and end up with the majority of the market profit share (49.2 % and 50.3%). Figure 5(middle) presents the profits and the price variations of three sellers over time, two of the sellers use minimax regret-based dynamic pricing and the other seller uses goal-directed pricing technique. For goal-directed strategy price computation, the parameter $daysInMarket$ is set to $1,000$ intervals and $initialInventory$ is set to $20,000$ units. The sellers using the minimax regret-based pricing technique are able to get higher shares of the profits, 35% and 42%, in Figure 5(middle), while the seller using goal-directed strategy gets about 23% of the total market profit share. Our simulations show that even in the market in which a seller has a limited supply of products, minimax regret-based pricing can outperform the

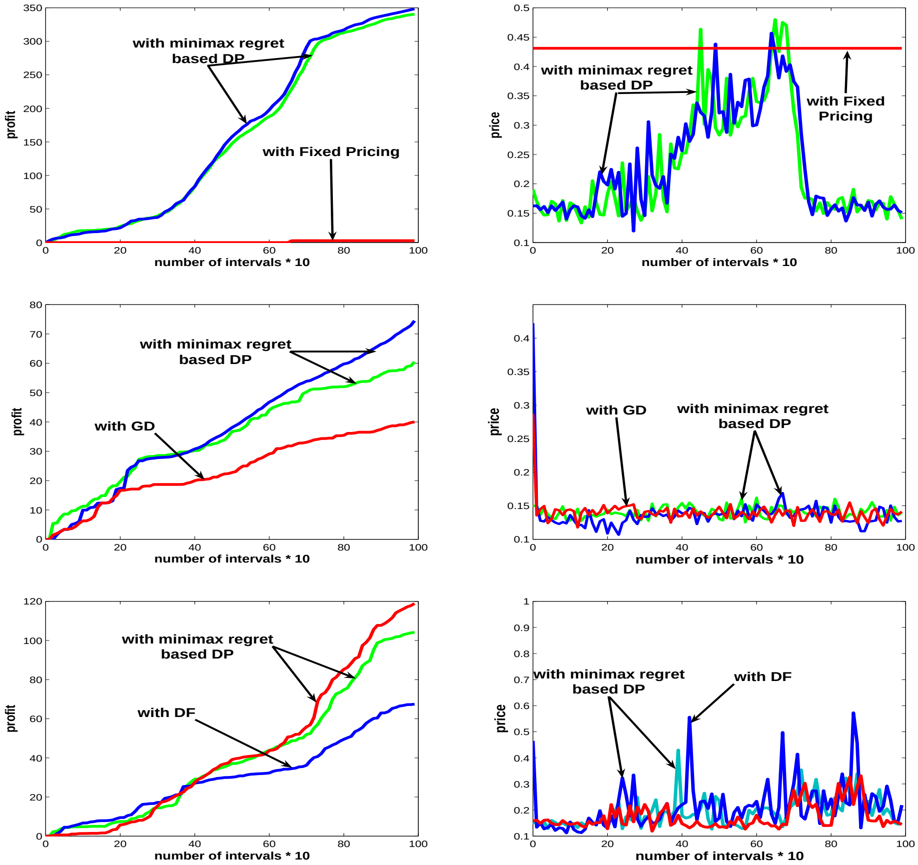

Fig. 5. Profit profile and the price competition between 3 sellers in the market with 500 buyers using Fixed Pricing or Minimax Regret (top), Goal-Directed (GD) or Minimax Regret (middle) and Derivative Follower (DF) or Minimax Regret (bottom) techniques.

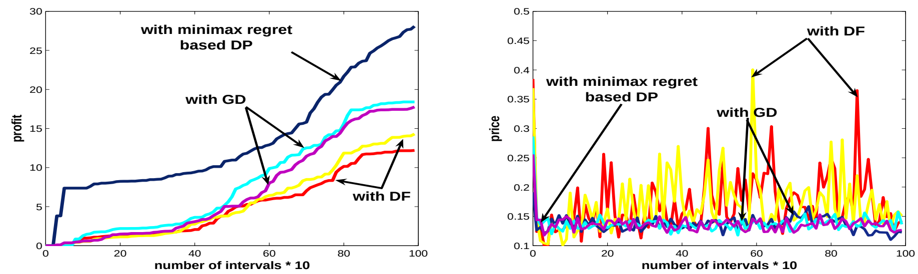

Fig. 6. Profit and price profiles of 5 sellers in the market with 1000 buyers using Goal-Directed (GD), Derivative Follower (DF) or Minimax Regret-based Dynamic Pricing techniques.

goal-directed strategy. Finally, Figure 5(bottom) illustrates the profit and price profiles of three sellers that use either the derivative-follower or the minimax regret-based dynamic pricing strategies. The market prices fluctuate consistently due to the competition between sellers. The seller using the derivative-follower pricing strategy ends up with the smallest market profit share (about 23% of the total profit in the market). The sellers that use the minimax regret-based dynamic pricing get the larger share of the market profit share (about 38% and 39%). The sellers that use the minimax regret for attribute prediction adjust to the buyers preferences and other market changes to obtain more profits.

Figure 6 illustrates the profit and price competition in a market with 1000 buyers and 5 sellers where 1 of the sellers use the minimax regret-based strategy and each of the remaining sellers use the goal-directed or derivative-following strategy. The two goal-directed sellers end up receiving 20% and 19% of the market's profits respectively. This is because the goal directed strategy is unable to reduce the posted price in response to price cuts by the other sellers because the strategy is dependent on the surplus inventory at the end of each interval. The derivative-follower seller raises prices until its profit goes down and then significantly lowers the price. The price adjustments by the derivative-follower are not very effective when the sellers using goal-directed and minimax regret-based pricing are simultaneously in the market with it. Consequently, the two sellers using derivative follower strategy for dynamic pricing receive 13% and 16% of the profits in the market. As sellers compete for profit shares, the sellers using minimax regret-based pricing adjust their prices more accurately and capture 32% of the market profit.

5 Related Work

Over the past few years, several researchers have considered the problem of automated dynamic pricing by sellers using software agents called pricebots. Kephart and Greenwald [14] analyze various dynamic price strategies such as game-theoretic, myopically optimal, derivative-following strategy, and Q-learning price setting strategy. An extension of this work [11] describes a no-regret learning based technique for automated dynamic pricing by sellers. In these algorithms sellers are assumed to have prior knowledge of some market parameters such as reservation prices of buyers, the distribution of buyers under each attribute of the product, and prices or profits of competing sellers. However, in most real-life economies, knowledge of such market parameters is either unavailable or has to be learned by the seller in real time. In contrast, in this paper, we do not assume prior knowledge of the reservation prices of buyers, the distribution of buyers under different product attributes and competitors' price information. Moreover, the minimax regret based attribute prediction and dynamic pricing techniques presented in this paper can determine product prices for sellers when the preferences of buyers over different product attributes change dynamically with time. In [12, 4], the authors have also considered the problem of dynamic pricing products. However, their settings contain only one seller and the main problem considered is to determine the optimal bundle of products and the price of the bundle that maximizes the seller's profit. Preference elicitation of consumers in a market has also been an active research topic in the area of decision support systems[13]. Recently, Lahaie and Parkes [15] have developed techniques based on machine learning for preference elicitation from consumers. Bayesian methods for preference elicitation have been researched in [7] where preference uncertainty is probabilistically quantified. In contrast, in our work, sellers don't use a density over possible utility functions and dynamically calculate the distribution of buyers over different product attributes. In [6, 19], Boutilier *et al.* use minimax regret as a decision criterion in constraint-based decision making. We have used a similar minimax regret-based technique to enable a seller's price-bot learn the consumer's preferences over different attributes and dynamically update the product's posted price. In contrast to the work in [6], sellers in our model cannot explicitly send queries to consumers to elicit their preferences. We have therefore considered the price quote sent by a seller to the buyer as a bound query and used the purchase decision made by the buyer after receiving the seller's price quote as a response to the bound query. Conitzer [8] showed that single-peaked preferences can be elicited using comparison queries if prior knowledge of some preference order structure or the preferences of one agent exists. However, in our model sellers interact with buyers only while offering price quotes or receiving an affirmative purchase decision. Therefore, our model is not amenable to comparison queries.

6 Conclusion

In this paper, we have described an algorithm that can be used by sellers to determine temporally changing buyer preferences across multiple product attributes, and, to dynamically update the posted product prices in a competitive market without explicit knowledge of various market parameters. There are several directions we plan to expand this work in the future. First, we are interested in investigating dynamic pricing algorithms in markets where sellers have information, possibly partially, about competitors' prices. With partial information about competitors' prices a seller could possibly infer the reason for a negative purchase decision by a buyer more efficiently and improve its buyer attribute prediction as well as its dynamic pricing performance, resulting in improved profits. Secondly, we plan to investigate a scenario where buyers are able to exchange information about sellers with each other. This results in more informed purchase decisions between buyers and could even lead to collisions between buyers to affect the prices being charged by sellers. Dynamic pricing over multiple attributes by competing sellers in a market with limited information about market parameters is a relevant yet challenging problem and we envisage that appropriate solution techniques in this direction will result in improved success of e-commerce technologies.

References

1. Bizrate Inc, URL http://www.bizrate.com
2. MySimon, URL http://www.mysimon.com
3. PriceGrabber, URL http://www.pricegrabber.com
4. M-F. Balcan, A. Blum, "Approximation algorithms and online mechanisms for item pricing," Proc. ACM-EC, 2006, pp. 29-35.
5. D. Bergemann, D. Ozmen, "Optimal pricing with recommender systems," Proc. ACM-EC, 2006, pp. 43-51.
6. C. Boutilier, R. Patrascu, P. Poupart, D. Schuurmans, "Constraint-based optimization and utility elicitation using the minimax decision criterion," Artificial Intelligence, vol 170, no 8, 2006, pp. 686-713.
7. U. Chajewska, D. Koller, R. Parr, "Making rational decisions using adaptive utility elicitation," Proc. of the 17th National Conference on AI, 2006, pp. 363-369.
8. V. Conitzer, "Eliciting single-peaked preferences using comparison queries," Proc. AAMAS, 2007, pp. 408-415.
9. P. Dasgupta, Y. Hashimoto, "Multi-attribute Dynamic Pricing for Online Markets using Intelligent Agents," Proc. AAMAS, 2004, pp. 277-284.
10. J. M. DiMicco, A. Greenwald, P. Maes, "Dynamic Pricing Strategies under a Finite Time Horizon," Proc. ACM-EC, 2001, pp. 95-104.
11. A. Greenwald, J. Kephart, "Probabilistic Pricebots," Proc. of the 5th International Conference on Autonomous Agents, 2001, pp. 560-567.
12. V. Guruswami, J. Hartline, A. Karlin, D. Kempe, C. Kenyon, "On profit-maximizing envy-free pricing," Proc. 16th annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 1164 - 1173.
13. R. Keeney, H. Raiffa, "Decisions with Multiple Objectives," Cambridge University Press, 1993.
14. J. Kephart, J. Hanson, A. Greenwald, "Dynamic pricing by software agents," Computer Networks, vol 32, no 6, 2000, pp. 731-752.
15. S. Lahaie, D. Parkes, "Applying learning algorithms to preference elicitation," Proc. of the ACM Conference on Electronic Commerce, 2004, pp. 180-188.
16. N. Li, P. Zhang, "Consumer Online Shopping Attitudes and Behavior: An Assessment of Research," 8th Americas Conference on Information Systems, 2002.
17. U. Shardanand, P. Maes, "Social Information Filtering: Algorithms for Automating "Word of Mouth"," Proc. ACM Conference on Human Factors in Computing Systems, 1995, pp. 210-217.
18. Y. Wei, "Future Orientation, Chronological Age and Product Attributes Preference," PhD Thesis, Georgia State University, 2007.
19. T. Wang, C. Boutilier, "Incremental utility elicitation with the minimax regret decision criterion," Proc. IJCAI 2003, pp. 309-316.