Computer Networks 33 (2000) 775–788

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

A Web marketing system with automatic pricing

Naoki Abe *, Tomonari Kamba [1]

NEC C&C Media Research Laboratories and Human Media Research Laboratories, 4-1-1 Miyazaki, Miyamae-ku, Kawasaki 216-8555, Japan

Abstract

We propose a new scheme of 'automatic pricing' for digital contents, and describe an implemented system as well as concrete pricing algorithms for it. Automatic pricing refers to a methodology of automatically setting sales prices to optimal prices, based on past prices and sales. In particular, we consider the case in which automatic pricing is done in order to maximize the profit of an on-line marketing site. We describe a demo site for on-line marketing with automatic pricing, which we call 'digiprice'. We will also describe the concrete pricing algorithms we employ in digiprice, and report on preliminary performance evaluation experiments we conducted using simulated data. The results of experimentation verify that our methods are practical in terms of both the speed of convergence to the optimal price and computational efficiency.

Keywords: Electronic commerce; On-line marketing; Automatic pricing; Machine; Learning; On-line learning

1. Introduction

With the recent technological and social developments surrounding the World Wide Web, it is fast developing as a medium for commerce as well as a medium for information exchange. Electronic commerce not only helps lubricate economical activities, but is bringing about a fundamental change to the practice of commerce that we have been familiar with over many centuries. Shopping sites on the Internet began at first as a mere copy of the off-line shops that make sales with fixed prices, but soon auction sites such as ebay and priceline [5,10] became well-accepted, and now 'reverse auction' sites are gaining popularity. Such forms of commerce did exist before the emergence of the World Wide Web, but these on-line marketing sites are qualitatively different from their off-line counterparts, due to the substantially greater reach they enjoy. These on-line sites are also attracting attention as a medium for community formation.

Electronic commerce is affecting what is being sold in addition to the way it is sold. Consumers can now purchase pictures, music and software as digital contents by downloading them from on-line sites. For manufacturers, digital contents allow mass-production with virtually no additional cost. Furthermore, electronic commerce is affecting the mechanism of pricing, one of the fundamental elements of commerce. On-line auction and reverse auction sites, by their large audience size, are affecting the existing practices of pricing. The prices are more and more determined by the consumers, rather than the manufacturers. Conceptually, this property is shared by any capitalist economy, but the speed and the

* Corresponding author. E-mail: abe@ccm.cl.nec.co.jp
[1] kamba@hml.cl.nec.co.jp

1389-1286/00/$ – see front matter
PII: S1389-1286(00)00071-2

manner by which the consumers can affect pricing in electronic commerce make it qualitatively different.

In this paper, we propose a new scheme of 'automatic pricing' for digital contents, and describe an implemented system as well as concrete pricing algorithms. Automatic pricing refers to a methodology of automatically setting sales prices to optimal prices, based on past prices and sales. In particular, we consider the case in which automatic pricing is done in order to maximize the profit of an on-line marketing site. In what follows, we begin by giving an outline of a demo site for on-line marketing with automatic pricing, which we call 'digiprice'. We will then describe the concrete pricing algorithms we employ in digiprice, and report on performance evaluation experiments we conducted using simulated data.

2. Digiprice: a Web-marketing system with automatic pricing

Digiprice is an on-line shopping server equipped with an automatic pricing function. The electronic payment function has not yet been implemented in the system and the site is operating internally on an intranet. It is therefore not freely available on the Internet. Fig. 1 shows the overall system architecture. As shown in this figure, a seller is expected to specify conditions of sales for each item, such as the initial price and the minimum possible price, and the system automatically adjusts the sales price so as to maximize the sales revenue within those conditions. For this purpose, the system keeps the time of sales in addition to the quantity of sales in its database. This is because how the amount of sales per unit time changes as the price is adjusted plays a key role in automatic pricing. The system constantly calculates the current sales price for each item, based on the past records of prices and sales and the conditions specified by the seller.

Fig. 2 shows the top page of digiprice. Fig. 3 gives a page for a seller to register an item for on-line sales, and Fig. 4 shows an example page for a consumer to select an item, to purchase or to examine. As shown in Fig. 4, a consumer can check to see how the prices of a given content have changed in the past, and refer to comments made by other users. In particular, Fig. 5 depicts what can happen, when the user clicks on the 'check market' button on the page shown in Fig. 4. The past prices of the specified item are plotted over time in the graph. The users can, for example, view this graph, try and predict what will happen to the price in the near future, and make their decision on whether and when to buy a given item. Fig. 6 shows an example of a page that appears when a user has requested to purchase a certain item. Purchase of an item can mean different things depending on what is being bought. For example, if it is a digital picture that is being purchased, then it may mean downloading of a high-quality digital picture file. If it is music, then it may mean downloading music data, which can then be enjoyed on a publicly available software.

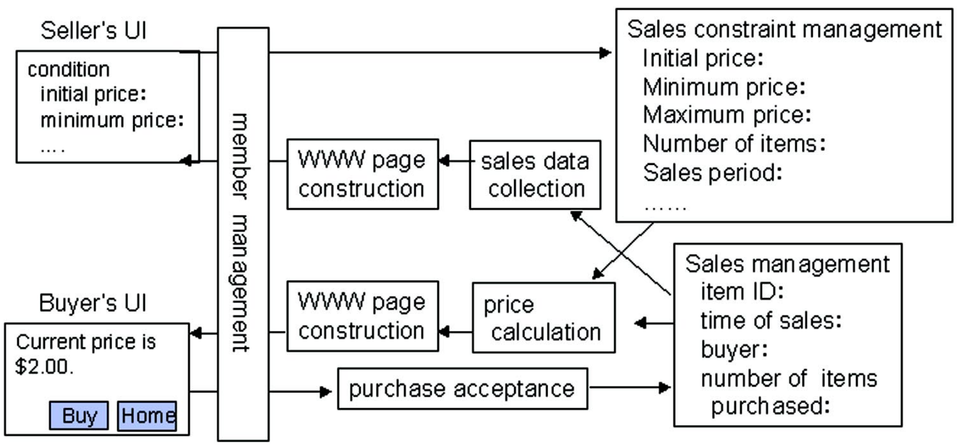

Fig. 1. System architecture of digiprice.

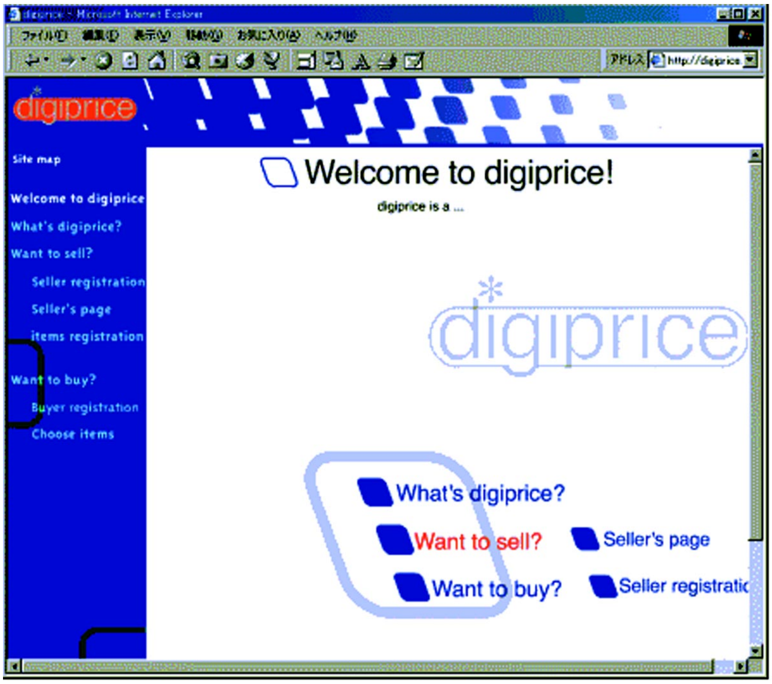

Fig. 2. Top page of digiprice.

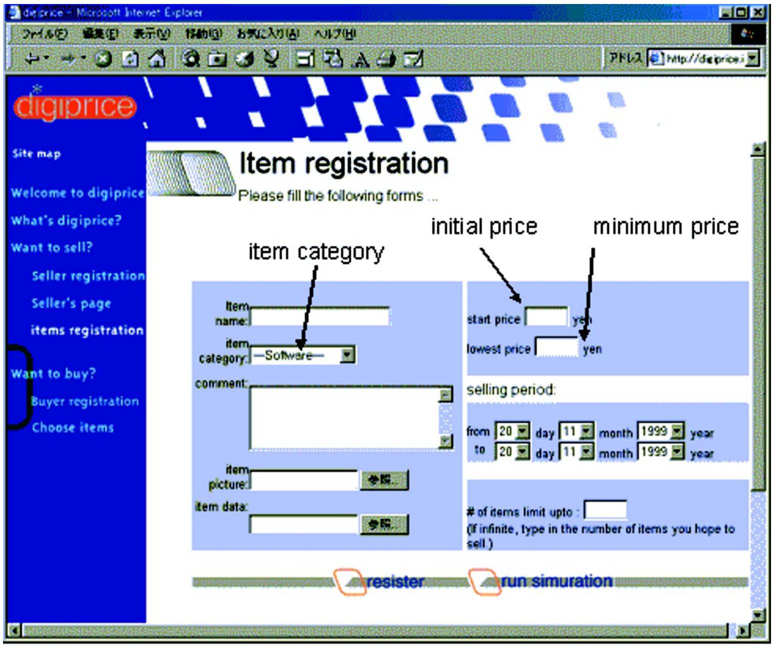

Fig. 3. Seller's page of digiprice.

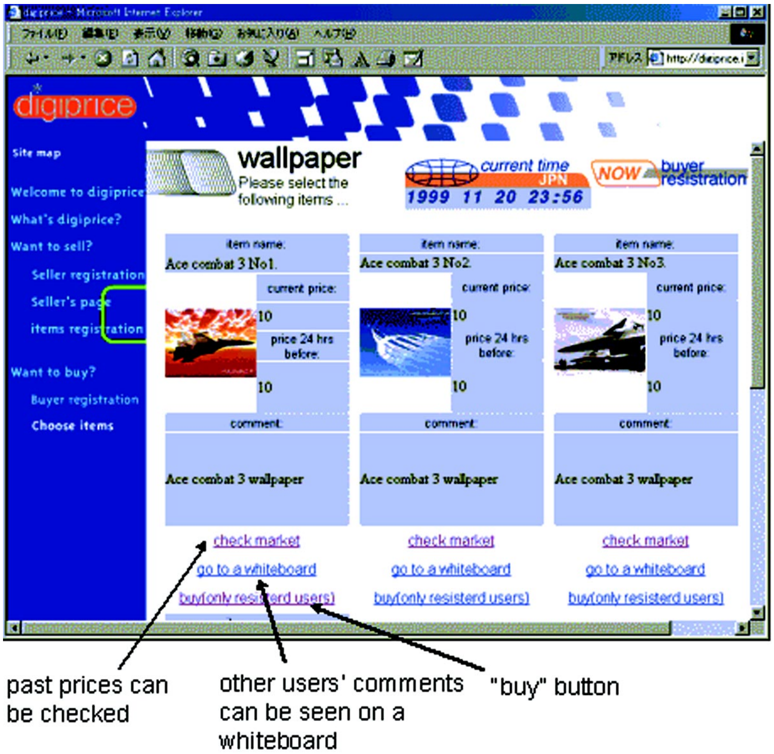

Fig. 4. User's page of digiprice.

Fig. 5. 'Check market' page of digiprice.

Fig. 6. Purchase page of digiprice.

3. Automatic pricing methods employed by digiprice

The automatic pricing methods that we propose and that are implemented as part of 'digiprice' are the following three.

(1) A pricing method that determines the price of each item independently, without using any attributes associated with them.
(2) A pricing method that determines the prices of all items at once, as a function of their attributes.
(3) A couple of selection methods that determine which items to display in order to optimize the trade-off between selling more immediately and estimating better the pricing function. This is to be used in combination with the above attribute-based method as the pricing method.

3.1. A pricing method for individual goods

With some exceptions, it is generally believed that the amount of commercial goods sold is a decreasing function of their prices. Let us denote the amount of sales per unit time of (a fixed item) by $S(p)$, at price p. Some goods are more sensitive to the price than others, and thus $S(p)$ is unknown a priori to an on-line Web marketing system. An especially desirable feature of on-line marketing is that it allows us to observe the amount of sales as a function of the price and optimize the price in an on-line fashion.

In general, profit per sales depends on the price as well as the amount of sales. In reality, this dependence is complicated due to factors such as economy to scale. Here, for simplicity, we suppose that the total profit can be approximated by the following function of price p and amount of sales $S(p)$, where we use $C(p, N)$ to denote the production cost (per item) at price p and amount of sales N.

$$P(p) = S(p) \cdot \big(p - C(p, S(p))\big).$$

We further assume that the total cost does not depend on the amount of sales, allowing us to simplify the above to:

$$P(p) = S(p) \cdot p - C.$$

This assumption is reasonable for digital contents.

Now the goal of an automatic pricing method is to find a price p that maximizes $P(p)$ quickly, and set the price accordingly. That is, it wishes to find p^* such that

$$p^* = \arg\max_p P(p).$$

We emphasize here that the goal is to quickly find p^*, and not necessarily to estimate the entire function $P(p)$.

Below, we will describe a pricing method that estimates p^* for each item independently, and sets the price automatically.

3.1.1. A pricing method based on stochastic approximation

Stochastic approximation is a general methodology for on-line function optimization, which estimates the maximum of a regression function by testing and obtaining the (estimate of) function values at points of its choice, and gradually converge to the optimum point (cf. [12]). In this section, we propose an automatic pricing method based on this general technique.

The basic idea of stochastic approximation is as follows. In trying to maximize $f(x)$, which dictates the expected reward at point x, it estimates the value of the derivative $f'(x)$ at its current point x, and determines the next point by updating the current position as $x + af'(x)$. Since we cannot observe the values of $f(x)$ directly, one need take care to ensure that the estimation of the derivative is credible. We basically follow the idea of the Kiefer–Wolfowitz method. Having set the initial point x to be some arbitrarily chosen point, this method repeats the following process.

(1) For step size Δ, appropriately determined as a decreasing function of the number of trials (up to that point), test at two points $x+\Delta$ and $x-\Delta$, and obtain rewards $\hat{f}(x-\Delta)$ and $\hat{f}(x+\Delta)$, whose expectations are dictated by the function f.

(2) Update x as follows:

$$x := x + a\frac{\hat{f}(x+\Delta) - \hat{f}(x-\Delta)}{\Delta}.$$

Here, in order to ensure that the x's so obtained converge to the (local) maximum point of f, certain constraints must be satisfied by the values of a and Δ, as will be specified later.

Based on this general idea, we propose a pricing method for a single good. As indicated above, we assume that the profit function can be approximated as

$$P(p) = S(p) \cdot p - C.$$

For maximization of $P(p)$, C can be ignored, so we further simplify the above to obtain

$$P(p) = S(p) \cdot p.$$

In other words, we are simply to maximize the total revenue.

There can be legal constraints on pricing (against dumping, for example), so we assume that there are maximum and minimum possible prices, $p_{\max}$ and $p_{\min}$, given a priori to the system. We assume also that the system is given a reasonable price p_{init} as the initial price. Based on these minimal pieces of information, our first pricing method (StochPrice: Stochastic Pricing) repeats the following process.

(1) Using step size Δ, determined as a decreasing function of the number of trials so far (for example, $\Delta = I^{-1/3}$, where I is the current trial number), conduct on-line sales at both prices $p + \Delta$ and $p - \Delta$ for a certain fixed period of time, and based on the amount of sales obtained during these periods, $S(p+\Delta)$ and $S(p-\Delta)$, calculate the profits (revenues) obtained for the respective prices:

$$P(p+\Delta) = S(p+\Delta) \cdot (p+\Delta),$$

$$P(p-\Delta) = S(p-\Delta) \cdot (p-\Delta).$$

(2) Update the current price p as follows:

$$p := p + \frac{A}{\Delta}\,\frac{P(p+\Delta) - P(p-\Delta)}{2T}.$$

Here, A is an update interval, set as a decreasing function of the trial number (for example $A = c \cdot I^{-1}$), and T is a measure of the duration of each trial. (To be precise, the size of T depends strongly on what time unit is used. In our experiments, we actually used the average number of visits to the site in a unit period as the value of T.)

(3) If p is either above $p_{\max}$ or below $p_{\min}$, then clamp its value so that it falls within these bounds.

In order to ensure convergence of the obtained prices to the (local) optimum, it suffices to see that the following conditions hold on A and Δ described above [12]:

$$\sum_{I=1}^{\infty} A(I) = \infty,$$

$$\sum_{I=1}^{\infty} \frac{A(I)^2}{\Delta(I)^2} < \infty.$$

Note that both of these conditions are met by the examples we gave earlier of $A = c \cdot I^{-1}$ and $\Delta = c \cdot I^{-1/3}$.

We give the details of this method as a pseudo-code below.

Algorithm: StochPrice
(p_{init}, $p_{\min}$, $p_{\max}$, T: Unit sales period)

1. Initialization
   1.1. Initial price $p := p_{\text{init}}$
   1.2. Trial number $I := 1$
2. **Repeat for** $I = 1$ until forever
   2.1. Set Δ as follows: $\Delta := I^{-1/3}$
   2.2. For a period of T, set the price to $p + \Delta$.
   2.3. Let $S(p+\Delta)$ be the amount of sales during this time.
   2.4. For a period of T, set the price to $p - \Delta$.
   2.5. Let $S(p-\Delta)$ be the amount of sales during this time.

2.6. Calculate the obtained profit as follows:

$$P(p+\Delta) = S(p+\Delta)\cdot(p+\Delta)$$

$$P(p-\Delta) = S(p-\Delta)\cdot(p-\Delta)$$

2.7. Set the update interval A as follows:

$$A := \frac{1}{I}$$

2.8. Update the current price as follows:

$$p := p + \frac{AP(p+\Delta) - P(p-\Delta)}{2T\Delta}$$

2.9. If necessary, clamp the value of p between the maximum and minimum possible prices.

$$p := \min\{p_{\max} - \Delta, \max\{p_{\min} + \Delta, p\}\}$$

3.2. Attribute-based automatic pricing method

The pricing method proposed in the previous section suffers from the shortcoming that each new product will basically have to be priced from scratch. Using attributes associated with goods and users, it is supposed that earlier experiences can be used to price new items more intelligently. In this section, we propose such a method.

Let us write X for a (binary) attribute vector associated with a fixed item, and write x_i for its ith component. These attributes can be attributes of the good, such as its category, those of a user, such as their demographic attributes, and their combinations. For example, based on attributes like $y_1 = 1$ iff cosmetics, $y_2 = 1$ iff woman, combined attributes like $x_1 = y_1 \cdot y_2$ can be obtained. Strictly speaking, therefore, these attributes are not attributes of the goods alone, but are attributes of potential 'sales,' consisting of a user, a good and possibly the environment. In digiprice, we use binary attributes of the contents, those of the users, and their logical combinations (conjunctions).

The basic idea behind the pricing method we propose here is that the optimal price of a good can be approximated by a linear function of its attributes. That is, there is a weight vector W having the same dimensionality as their associated attribute vectors, such that for an arbitrary item, its profit function $P_X(p)$ approximately attains a maximum at $W \cdot X$, where X is its associated attribute vector, i.e.

$$p^*_X = \arg\max_p P(p) = W \cdot X.$$

Here, we emphasize that this assumption is much weaker than an assumption that $P_X(p)$ itself can be approximated by a simple function. In general, it is supposed that $P_X(p)$ will take a complicated form, but it is reasonable to suppose that its optimal price can be linearly approximated. Since the goal of an automatic pricing method is not necessarily to estimate $P_X(p)$ but that of p^*_X, this assumption helps us design a simple method for it.

3.2.1. Automatic pricing method based on linear approximation of optimal price

The pricing method we propose here is based on the Kiefer–Wolfowitz method as before, but here the search for optimality is done in a higher dimensional attribute space.

(1) For each item, calculate its associated vector based on its attributes and those of the current user.

(2) For each item i, set the initial price $p(i)$ to $W \cdot X(i)$, using the weight vector W.

(3) For each item i, generate a random vector $\vec{\Delta}(i)$ of length Δ. Here, Δ is a step size, determined as a decreasing function of the trial number I (for example, $\Delta = c \cdot I^{-1/3}$).

(4) Using the vector $\vec{\Delta}(i)$ thus obtained, set the current price for each item i as follows:

$$p(i) := (W + \vec{\Delta}(i)) \cdot X.$$

Here, for each item, if $p(i)$ goes above the maximum price or below the minimum price, then clamp the vector $\vec{\Delta}(i)$ so that this is not so.

(5) Conduct on-line sales at the above price for a certain period.

(6) Next, conduct on-line sales for the same period, with the price set as follows:

$$p(i) := (W - \vec{\Delta}(i)) \cdot X(i).$$

Here, too, $\vec{\Delta}(i)$ should be clamped if necessary.

(7) For each item, for the respective prices, calculate the total profits, based on the amount of sales obtained for each of the above periods, $S(W + \vec{\Delta}(i))$ and $S(W - \vec{\Delta}(i))$:

$$P(W+\vec{\Delta}(i)) = S(W+\vec{\Delta}(i)) \cdot X(i)(W+\vec{\Delta}(i))$$

$$P(W-\vec{\Delta}(i)) = S(W-\vec{\Delta}(i)) \cdot X(i)(W-\vec{\Delta}(i))$$

(8) Once for each i, update the weight W using the value of $\vec{\Delta}(i)$, as follows:

$$W := W + \frac{A\vec{\Delta}(i)}{|\vec{\Delta}(i)|} \frac{P(W + \vec{\Delta}(i)) - P(W - \vec{\Delta})(i)}{2T}$$

Again, A is an update interval set as a decreasing function of I (for example, $A = c \cdot I^{-1}$).

The details of this method, FeaturePrice(Feature-based Pricing) are shown as a pseudo-code below.

Algorithm: FeaturePrice
(W_{init}: initial weight vector; T: unit sales period)

1. Initialization
   1.1. Set weight vector $W := W_{\text{init}}$
2. **Repeat for** $I = 1$ until forever
   2.1. **For** item $i = 1$ until N (number of items)
      2.1.1. $X(i) :=$ attribute vector for item i
      2.1.2. $p(i) := W \cdot X(i)$
      2.1.3. $\Delta := I^{-1/3}$
      2.1.4. $\vec{V}(i) =$ Random-Vector()
      2.1.5. $\vec{\Delta}(i) = \Delta \cdot (\vec{V}(i))/(|\vec{V}(i)|)$
   2.2. For a time period of T, set the price of each item (i) to $p(i) := (W + \vec{\Delta}(i)) \cdot X(i)$ and conduct sales.
   2.3. Let $S(W + \vec{\Delta}(i))$ be the amount of sales thus obtained for each item (i).
   2.4. For a time period of T, set the price of each item (i) to $p(i) := (W + \vec{\Delta}(i)) \cdot X(i)$ and conduct sales.
   2.5. Let $S(W - \vec{\Delta}(i))$ be the amount of sales thus obtained for each item (i).
   2.6. For each item i, calculate the revenue raised during this time.

$$P(W + \vec{\Delta}(i)) = S(W + \vec{\Delta}(i)) \cdot X(i)(W + \vec{\Delta}(i))$$

$$P(W - \vec{\Delta}(i)) = S(W - \vec{\Delta}(i)) \cdot X(i)(W - \vec{\Delta}(i))$$

   2.7. **For** $i = 1$ until number of items
   Update the weight vector W as follows:

$$W := W + \frac{A}{\Delta} \frac{P(W + \vec{\Delta}(i)) - P(W - \vec{\Delta})(i)}{2T}$$

3.3. Methods for selecting goods to display

We have so far discussed how to automatically set the prices of goods for on-line marketing, in order to maximize the resulting revenue. When the number of items to be sold at a particular site is large enough, however, there is the additional issue of which items to 'display on the show window'. Even if, in principle, the number of items that can be displayed on an on-line marketing site is unlimited, the degree of exposure to the user is heavily influenced by whether they are displayed on the top page, etc. In this section, we consider how to optimize both the prices and the selection of items to display, in order to maximize the total revenue obtained at an on-line marketing site.

At the heart of the issue raised here, is the trade-off known as the 'Exploration–Exploitation trade-off' in the literature on on-line learning (reinforcement learning in particular [7]), formulated below for the current problem of our concern.

(1) If one wishes to maximize the immediate revenue, one should select those items estimated to have the maximum expected revenue.
(2) If one wishes to maximize the total revenue obtained in the long run, one should also take care to display a variety of items, so that their optimal prices, and hence their expected maximum revenues, can be reliably estimated.

The issue of the Exploration–Exploitation trade-off has been addressed by a number of authors in the literature [1–3,6,8], but the work in [1,8], in particular, does so in the context of on-line learning of (probabilistic) linear functions, and is closely related to the problem studied here.

In trying to resolve this trade-off, we consider the following three measures.

(1) *The immediate pay-off of each item*. In particular, we use the profit obtained during the last sales period ($2T$) for each item, i.e.:

$$\text{PTotal}(i, W) = (P(W + \vec{\Delta}(i)) + P(W - \vec{\Delta}(i)).$$

(2) *Variety of attribute vectors*. We use the sum of Hamming distances among the vectors in the set of items selected to be displayed, i.e.:

$$H(S) = \sum_{u,v \in S} \sum_i |X(u)_i - X(v)_i|.$$

(3) *Uncertainty of estimating the optimal price function*. We use the difference between the revenues obtained in the first half and the second half of the last sales period:

$$\mathrm{PDiff}(i, W) = |P(W + \vec{\Delta}(i)) - P(W - \vec{\Delta}(i))|.$$

By combining (1) and (2), and (1) and (3) of these three measures, we propose two methods for selecting the items to display.

(1) *Variety Selection*. From among the set of candidate items, select a set of items such that the weighted sum of the variety measure and the immediate pay-off measure is maximized, that is, select S such that

$$\sum_{i \in S} \lambda_1 \mathrm{PTotal}(i, W) + \lambda_2 H(S).$$

Here, λ_1 and λ_2 are parameters controlling the relative contributions of the two measures.

(2) *Uncertainty Selection*. From among the set of candidate items, select the top N items, maximizing the sum of the immediate pay-off and the uncertainty measure:

$$\begin{aligned} &\mathrm{PTotal}(i, W) + \mathrm{PDiff}(i, W) \\ &\quad = P(W + \vec{\Delta}(i)) + P(W - \vec{\Delta}(i)) \\ &\qquad + |P(W + \vec{\Delta}(i)) - P(W - \vec{\Delta}(i))| \\ &\quad = 2 \cdot \max\{P(W + \vec{\Delta}(i)), P(W - \vec{\Delta}(i))\}. \end{aligned}$$

Thus, this method coincides with that of maximizing the larger of the revenues raised for the two different prices.

Since Variety Selection is a strategy that is more oriented towards exploration, we generally switch from Variety Selection to the method of selecting those expected to bring most immediate revenues, after an initial 'exploration' phase. For Variety Selection, strict maximization of $\sum_{i \in S} \lambda_1 \mathrm{PTotal}(i, W) + \lambda_2 H(S)$ would be computationally infeasible. Thus, we settle with the following heuristic. We start with the set consisting with the top N items in terms of the immediate pay-off, and then repeatedly perform random swaps, if such a swap results in increasing the value of the above objective function.

Algorithm: Variety Selection
(W: current weight vector; G: set of items; n: number of items to be displayed; N: number of iterations)

1. Initialization
   1.1. Sort G in increasing order of $\mathrm{PTotal}(i, W)$.
   1.2. $S := \mathrm{First} - n(G, n)$
   1.3. $\overline{S} := \backslash S$
2. **Repeat for** $i = 1$ until N
   2.1. Randomly select item $j \in \overline{S}$.
   2.2. If there exists item k such that exchanging j, k would result in increasing the value of

   $$\sum_{i \in S} \lambda_1 \mathrm{PTotal}(i, W) + \lambda_2 H(S)$$

   then make that exchange and update S and $\overline{S}$.
3. Output S.

We refer to the resulting on-line marketing strategy, consisting of good pricing and selection, by such names as StochPrice (Uncertainty) and FeaturePrice (Variety).

4. Performance evaluation

4.1. Experimental procedure

4.1.1. Consumption model

We model the consumption behavior of users by the following stochastic process.

At each time t and price p, do the following:

(1) Determine the number of visits $V(t)$ in a unit time interval to the marketing site in question, according to a normal distribution with a fixed average and variance.

(2) Determine the probability of purchase by the following function of time t since the beginning of sales and price p:

$$p(t, p) := F(t) \cdot G(p)$$

where the exact forms of $F(t)$ and $G(p)$ will each take one of three possible forms (to be described shortly). The idea is that the purchase probability is governed by a time-dependent factor and a price-dependent factor, which are relatively independent of one another.

(3) Determine the number of sales $S(t, p)$ in a given time period T by:

$$S(t, p) := V(t) \cdot p(t, p).$$

We considered the following three forms for $F(t)$.

(1) $F(t)$ is constant.

$$F(t) = C$$

(2) $F(t)$ decreases exponentially in time.

$$F(t) = C\ \exp\{-Kt\}$$

(3) $F(t)$ is normally distributed.

$$F(t) = C\ \exp\{-K(t-c)^2\}$$

For $G(p)$, we assume one of the following three forms.

(1) $G(p)$ is constant.

$$G(p) = C$$

(2) $G(p)$ decreases exponentially in price.

$$G(p) = C\ \exp\{-Kp\}$$

(3) $G(p)$ is a logistic function (smooth step function).

$$G(p) = C\frac{1}{1+\exp\{K(p-c)\}}$$

For most of the experimental results we reported on, we used the first choice for $F(t)$ and the last choice for $G(p)$.

4.1.2. Generation of attributes

The attributes associated with goods are generated by either of the following two methods: the independent method, and the random walk method. In the independent method, we generate each attribute vector by randomly generating a bit (0 or 1) by a fair coin. In the random walk method, we first generate a random attribute vector as above. We then generate the next vector by probabilistically flipping each bit independently, with a small fixed probability. This process is repeated until the desired number of vectors are obtained.

We then use these attributes to determine the consumption model for each item, so that items having similar attribute vectors tend to have similar consumption patterns. More concretely, we determine each of the constants (such as C, K, c) determining $F(t)$ and $G(p)$ as the product of the attribute vector and a real valued vector, which is randomly generated for each run.

4.1.3. Site visits

The number of site visits $V(t)$ is determined independently for each item (by a normal distribution), when there is no issue of selecting items to display. When a subset of items is to be selected for display, the number of site visits for the selected items, and those for the items not selected, are determined by two different normal distributions. We ensure that the distribution of $V(t)$ for the selected items has a much higher mean than for the non-selected ones.

4.2. Experimental results

For all cases we consider, we basically used the total (cumulative) revenues obtained, for a fixed item or an ensemble of items, as the performance measure for our pricing methods. In most cases. we compared the performance of the proposed pricing methods against that of the method of just keeping the initial price, and the ideal method of using a near optimal price from the beginning. Since the ‘optimal price’ was not available in closed form for most of our experiments, in our plots, we substituted the optimal price by the estimated optimal price, as obtained by our pricing methods. The performance plots are averaged over five randomized runs, unless otherwise noted.

4.2.1. Results on StochPrice

Fig. 7 plot the cumulative and instantaneous (per unit sales period) revenues obtained by the following three methods as a function of the number of trials: (1) keeping the initial price; (2) using StochPrice; and (3) using a near-optimal price from the beginning. In this experiment, various parameters of StochPrice were set as follows. The constants within F and G, which determine the purchase probability, were set as $C = 1.0$, $K = 0.5$, and $c = 20$. This means that the purchase probability is at most 0.5 and is around 0.1 for prices that are two units higher than the optimal price (which is around 16). The number of site visits per unit time, $V(t)$, was set to be distributed around 300 visits. Thus, at each unit sales period, a typical number of sales is in the tens.

Note that the third method is an ideal and impractical method, since the near optimal price is not available a priori. It is clearly seen in these graphs that after suffering a small initial loss due to the ne-

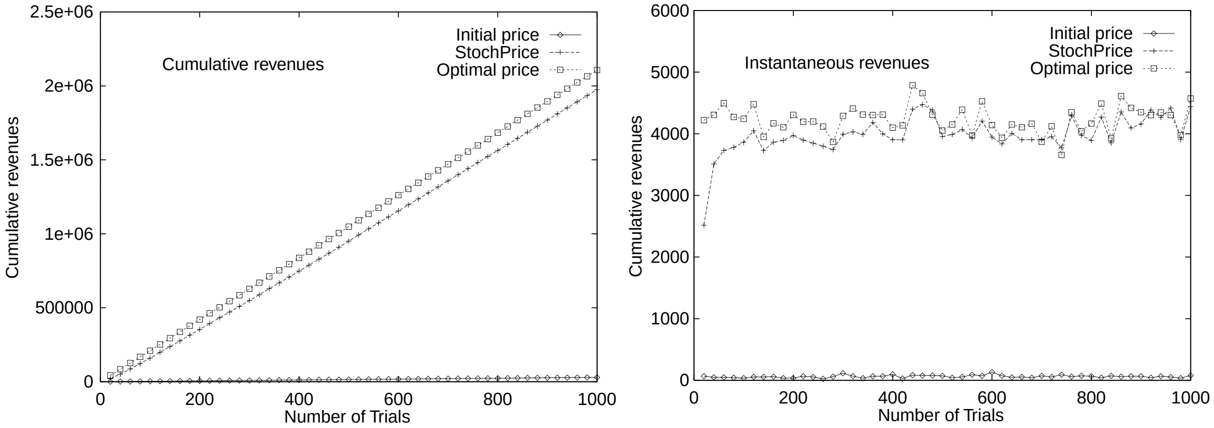

Fig. 7. Cumulative revenues (left) and instantaneous revenues (right) obtained by (1) keeping the initial price, (2) using StochPrice, and (3) using a near-optimal price from the beginning.

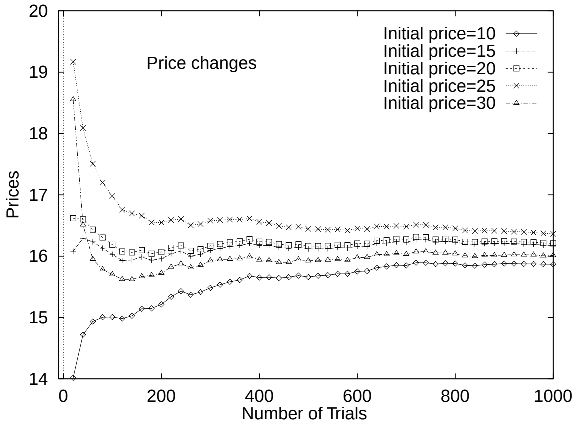

Fig. 8. Price changes made by StochPrice over 1000 trials as a function of the initial price.

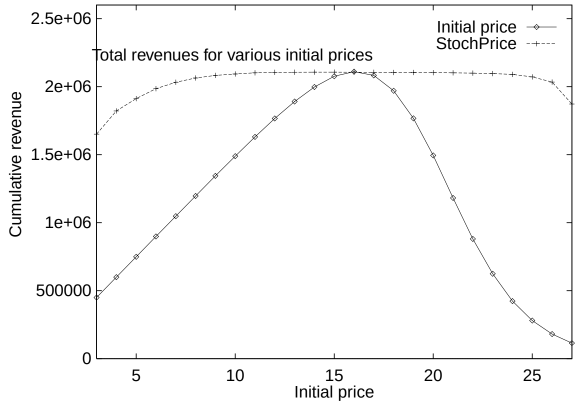

Fig. 9. Cumulative revenues obtained by StochPrice and by keeping the initial price, as a function of the initial price.

cessity to learn the optimal price, StochPrice quickly catches up with the ideal method in its performance.

Fig. 8 shows how the price of a particular item changes over time (on a particular run) using StochPrice, for a variety of choices of the initial price. It is seen that the optimal price is approximately 16 dollars, and whatever the initial price is (within the range shown here), they converge to the optimal price. Fig. 9 plots how the cumulative revenue obtained by StochPrice over 1000 trials changes as a function of the initial price. In the graph, this is compared with how the revenue changes if one kept the initial price, also as a function of the initial price. A rather dramatic difference is observed: when the initial price is kept, the total revenue quickly falls off when the initial price is wrongly set, but using StochPrice, one can see that the total revenue is insensitive to the choice of the initial price for a good range of initial prices.

4.2.2. Results on FeaturePrice

As before, we plot the cumulative and instantaneous (per unit sales period) revenues obtained by the following methods: (1) keeping the initial price, (2) using FeaturePrice, and (3) using a near optimal price from the beginning (see Fig. 10). In this experiment, the number of items was set to be 30, and the number of attributes was 10. Other parameters

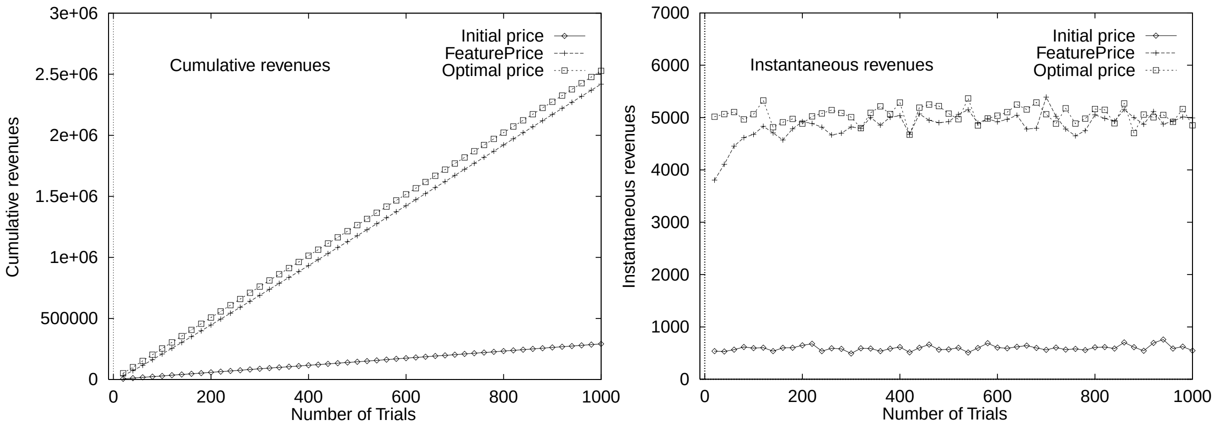

Fig. 10. Cumulative revenues (left) and instantaneous revenues (right) obtained by (1) keeping the initial price, (2) using FeaturePrice, and (3) using a near-optimal price from the beginning.

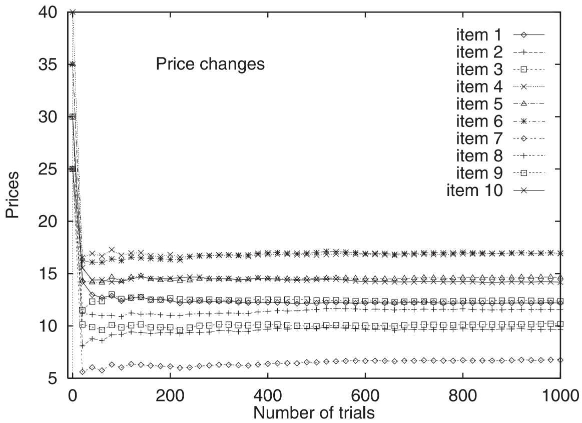

Fig. 11. Price changes made by FeaturePrice for 10 out of the 30 items, over 1000 trials.

controlling F and G were set as follows: $C = 1.0$, $K = 0.9$, and $c = 0.37$. $V(t)$ was set to be averaged at 300 as before.

It is clearly seen that FeaturePrice also improves the revenue of a marketing site significantly. Whether the prices computed by FeaturePrice converge to the respective optimal price is harder to see, but we plotted how the prices are changed by FeaturePrice: Fig. 11 exhibits how the prices are changed by FeaturePrice for 10 (chosen unintentionally) out of the 30 items used in the above experiment. One can see here that although the pricing is done via learning of a single target vector, a great variety of pricing is realized by utilizing the attributes attached to the items being sold. We plan in the near future to compare the performance of FeaturePrice with that of StochPrice in realistic settings, which we will report on in a full paper.

4.2.3. Results on Good Selection methods

We compared the revenues obtained by the two methods we propose to address the Exploration–Exploitation trade-off, namely Uncertainty Selection and Variety Selection, with the method of always selecting those items that are estimated to bring about the maximum revenue (MaxProfitSelection). Here, the exploration period for Variety Selection was set at 200 out of the 1000 trials in total. The parameters controlling F and G and $V(t)$ were set identically to the experiment for FeaturePrice.

Fig. 12 shows the cumulative and instantaneous revenues obtained by each of the three methods. It is seen that the revenues obtained by both Uncertainty Selection and Variety Selection significantly out-perform that of MaxProfit, with Uncertainty Selection being the favored one for the experimental conditions we tried. Since Uncertainty Selection is computationally cheaper than Variety Selection, it appears to be the method of choice.

5. Concluding remarks

We have proposed a new concept of on-line marketing with automatic pricing. In particular, we ex-

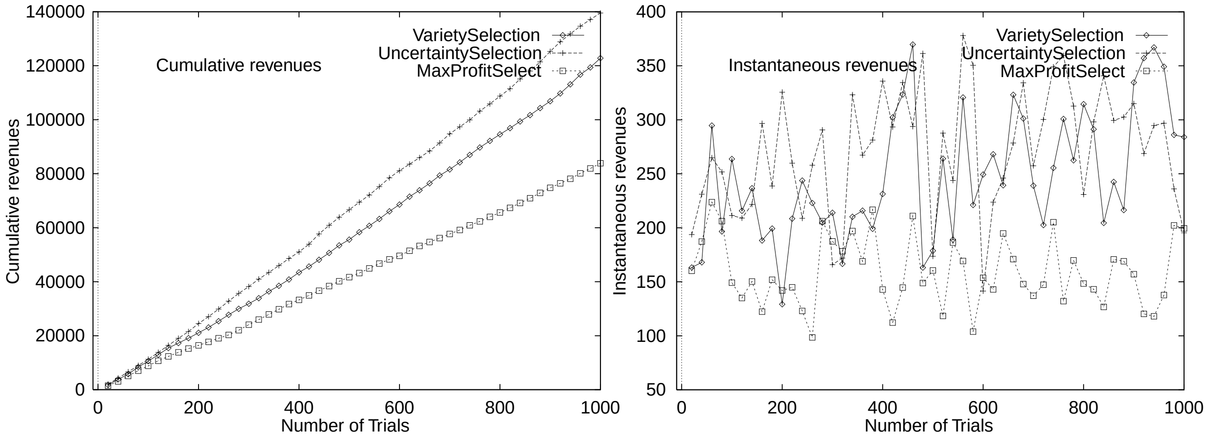

Fig. 12. Cumulative revenues (left) and instantaneous revenues (right) obtained by VarietySelection, UncertaintySelection and MaxProfit-Selection.

hibited an implemented system and concrete pricing algorithms designed to maximize the profit of the sellers. The proposed scheme of automatic pricing is distinguished from other forms of dynamic pricing that are currently employed by on-line marketing sites on the Internet. Most Web-marketing sites with dynamic pricing schemes base their pricing on the demand–supply information that is available via the Internet. Auction [5,10] and reverse auction sites [9], as well as other pricing information services [4] share the property that the pricing is determined as a result of competition between buyers and sellers. Among those Web-marketing sites that are currently operating on the Internet, perhaps ‘OutletZoo’ [11] employs a pricing scheme that is most closely related to ours. In particular, OutletZoo offers a dynamic pricing scheme they call ‘Automatic Drop’. It is a means to ensure that excess merchandises that manufacturers wish to sell are eventually all sold, and it works by simply making “prices fall at a seller determined percentage at regular intervals until everything is sold”. A variation of our automatic pricing scheme which is sensitive to constraints posed by the amount of goods in stock may prove to be a viable alternative. In the future, we wish to investigate extensions of the proposed pricing scheme to other scenarios and purposes, such as in a two-way interactive marketing environment like auction or in the presence of physical stock constraints. We believe that, through such extensions and variations, the concept of automatic pricing we propose in this paper can be developed into a model of on-line marketing that can truly benefit the buyers as well as the sellers.

Acknowledgements

We would like to thank Phil Long of National University of Singapore for valuable discussions on topics related to this paper. We also thank the anonymous referees for providing invaluable information. We wish to thank Dr. Doi, Dr. Koseki, Dr. Koike, Dr. Sakata, and Dr. Goto of NEC for their encouragement for this research. Finally, we thank Mr. Omoto and Dr. Takada of NIS for their programming efforts.

References

[1] N. Abe and P. Long, Associative reinforcement learning using linear probabilistic concepts, in: Proc. 16th International Conference on Machine Learning, 1999, pp. 3–11.

[2] N. Abe and J. Takeuchi, The ‘lob-pass’ problem and an on-line learning model of rational choice, in: Proc. 6th Annual ACM Conference on Computational Learning Theory, 1993, pp. 422–428, 1994, pp. 1198–1204.

[3] D.A. Berry and B. Fristedt, Bandit Problems, Chapman and Hall, New York, 1985.

[4] Dealtime, An Internet marketing service site, http://dealtime.com/.

[5] Ebay, An Internet auction site, http://www.ebay.com/.

[6] D.P. Helmbold, N. Littlestone and P.M. Long, Apple tasting and nearly one-sided learning, in: Proc. 33rd Annual Symposium on the Foundations of Computer Science, 1992.

[7] L. Kaelbling, Associative reinforcement learning: functions in k-DNF, Machine Learning 15 (3) (1994) 279–298.
[8] P.M. Long, On-line evaluation and prediction using linear functions, in: Proc. 10th Annual Conference on Computational Learning Theory, 1997, pp. 21–31.
[9] Nextag, An Internet reverse auction site, http://nextag.com/.
[10] Priceline, An Internet auction site, http://www.priceline.com/.
[11] OutletZoo, An Internet marketing site, http://outletzoo.com/.
[12] M.T. Wasan, Stochastic Approximation, Cambridge University Press, Cambridge, 1969.

Naoki Abe received his B.S. and M.S. degrees from Massachusetts Institute of Technology in 1984, and his Ph.D. degree from the University of Pennsylvania in 1989, all in computer science. After holding a postdoctoral researcher position at the University of California, Santa Cruz, he joined the NEC Corporation in 1990, where he is currently principal researcher in the C&C Media Research Laboratories. He is also a visiting associate professor in the department of computational intelligence and systems science of Tokyo Institute of Technology. His research interests include theories and applications of machine learning to various domains, including Internet information mining and navigation.

Tomonari Kamba received his B.E., M.E. and Ph.D. in electronics from the University of Tokyo in 1984, 1986 and 1997, respectively. He joined the NEC Corporation in 1986, and he has been engaged in user interface design methodology, multimedia user interface, software agent, and Internet information service technology. He was a visiting scientist at the Graphics, Visualization and Usability Center at the College of Computing, Georgia Institute of Technology, from 1994 to 1995. He is now research manager in the Human Media Research Laboratories. Dr. Kamba is a member of ACM SIGCHI and the Information Processing Society of Japan.