Reinforcement Learning Applications in Dynamic Pricing of Retail Markets

C.V.L. Raju
Department of C.S.A
Indian Institute of Science
Bangalore-560012, India
raju@csa.iisc.ernet.in

Y. Narahari
Department of C.S.A
Indian Institute of Science
Bangalore-560012, India
hari@csa.iisc.ernet.in

K. Ravikumar
IBM India Research Laboratory
Indian Institute of Technology
New Delhi-110016, India
rkaruman@in.ibm.com

Abstract

In this paper, we investigate the use of reinforcement learning (RL) techniques to the problem of determining dynamic prices in an electronic retail market. As representative models, we consider a single seller market and a two seller market, and formulate the dynamic pricing problem in a setting that easily generalizes to markets with more than two sellers. We first formulate the single seller dynamic pricing problem in the RL framework and solve the problem using the Q-learning algorithm through simulation. Next we model the two seller dynamic pricing problem as a Markovian game and formulate the problem in the RL framework. We solve this problem using actor-critic algorithms through simulation. We believe our approach to solving these problems is a promising way of setting dynamic prices in multi-agent environments. We illustrate the methodology with two illustrative examples of typical retail markets.

1 Introduction

Sellers have always faced the problem of setting the right prices for goods and services that would generate the maximum revenue for them. The advent of the Internet and electronic commerce has led to pricing of goods and services to move from from a fixed pricing model to a dynamic pricing model. According to Bichler *et al* [1], there are two important reasons for this shift from fixed pricing to dynamic pricing: (1) reduction of transaction costs associated with dynamic pricing by eliminating the need for buyers and sellers to be physically present in time and space to participate in markets; (2) the Internet has increased the number of customers and amount of competition thus leading to price uncertainty and demand volatility. Businesses and retailers have now realized that using a single fixed price in volatile Internet markets can be inefficient and ineffective.

The common problem to any seller in an electronic retail market is that, given a product, how should one vary the price over time and how much discrimination is to be given to different customers. In this paper we discuss how sellers can use algorithmic methods or automated pricing agents (also called price bots [7]), to determine optimal dynamic prices that maximize their revenue. In particular, we investigate the use of reinforcement learning (RL) techniques and algorithms such as Q-learning [18], and actor-critic algorithms [8] in solving dynamic pricing problems.

1.1 Related Work

The use of learning-based models in solving dynamic pricing problems is relatively recent. Lawrence [9] considers the problem of pricing a bid by a seller in a multi-seller procurement situation. Machine learning is used to learn directly the probability of winning from a database of bid transactions with known outcomes.

Ravikumar, Gupta, and Kumar [5] consider a web-based multi-unit Dutch auction where the auctioneer progressively decrements per unit price of the items and model the problem of finding a decrementing sequence of prices so as to maximize total expected revenue, in the presence of uncertainty with regard to arrival pattern of bidders and their individual price-demand curves. The above decision problem is modeled as a single agent RL in an uncertain non-stationary auction environment. Under the assumption of independent bidder valuations, the authors develop a finite horizon Markov decision process model with undiscounted returns and solve it using a Q-learning algorithm.

Ravikumar, Saluja, and Batra [12] study a service market environment with two sellers who compete to service a stream of buyers who are of two varieties, informed and uninformed. They assume that both the sellers follow an RL-based adaptive behavior and model the system as a general sum Markovian game. They propose an actor-critic type of RL scheme (a variant of the scheme proposed by Konda and Borkar [8] and provide experimental results on convergence.

Proceedings of the IEEE International Conference on E-Commerce (CEC'03)
0-7695-1969-5/03 $17.00 © 2003 IEEE

1.2 Contributions

In this paper, we investigate the dynamic pricing problem in a typical retail market where there is a single seller or multiple (competing) sellers. The customers (or buyers) are distinguished into two categories, shoppers and captives, following the model of Varian [17]. Shoppers go after volume discounts whereas captives do not base their decisions on considerations such as volume discounts. For example, a captive may typically buy one item at a item without worrying about the price. The decision problem here is to determine the optimal prices to be chosen by the seller(s) so as to maximize revenue, under the assumed behavior of the customers and the environment.

First, we consider a single-seller market and model the dynamic pricing problem as a Markov decision process. Since the seller does not know the environment dynamics completely, we find it natural to use Q-learning which is an RL-based procedure for solving Markovian decision problems.

Next, we consider a two-seller market and model the dynamic pricing problem as a two-person stochastic game, where both the players (that is, sellers) follow RL-based adaptive behavior. Q-learning based multi-agent algorithms have been suggested in [6, 10, 11]. We consider a general-sum Markovian game and use an actor-critic-type of RL scheme such as proposed in [8, 3, 12]. Like in [12], we model the two players as two actor-critic learners, with each player associated with one pair of actor (policy) and critic (policy evaluation). We adopt the actors on different time scales with the intuition that, the slower player sees the other player to be in equilibrium and the faster player sees the other player as quasi-static and hence, both the game might likely converge to a Nash equilibrium. Different actors at different time scales is a reasonable model when the players' capabilities in acquiring information are different. By a simulation study, we show how a player (seller) learns his optimal policy in such a dynamic pricing game of two-players (two competing sellers) of a retail market.

The rest of the paper is organized as follows. Section 2 provides some background on reinforcement learning, particularly on Q-learning and actor-critic algorithms. Section 3 presents the single-seller dynamic pricing model, application of Q-learning algorithm, and simulation results. Section 4 presents the two-seller dynamic pricing model, application of two actor-critic learners, and simulation results.

2 Reinforcement Learning: An Overview

The term *reinforcement learning* (RL) originates in studies of learning behavior of animals. Mathematically, it falls somewhere between the supervised and unsupervised learning paradigms of pattern recognition. RL neither calls for exact information about error from the environment, nor works with no information from the environment. RL expects a "reinforcement signal" from the environment indicating whether or not the latest move is in the right direction. RL procedures have been established as powerful and practical methods for solving Markovian Decision Problems (MDP).

2.1 Markov Decision Process (MDP)

Consider a process, observed at time epochs *t* = 0, 1, ..., to be in one of the states *i* ∈ S. After observing the state of the process an action $a \in A = \{a^1, a^2, \ldots, a^m\}$ is taken, where A is the set of all possible actions. If the process is in state *i* at time *n* and action *a* is chosen, then two things occur, (1) we incur a cost/reward R(*i,a*) (2) the next state of the system is chosen according to the transition probabilities $P_{ij}(a)$. If we let X_n denote the process at time n and a_n the action chosen at that time, then assumption (2) can be stated as:

$$P_{ij}(a) = P(X_{n+1} = j \mid X_0, a_0, X_1, a_1, \ldots, X_n = i, a_n = a) \quad (1)$$

We suppose that we always work with the case where $\mid R(i,a) \mid < M \ \forall$ *i,a*. A policy is any rule for choosing actions. An important subclass of policies is the class of stationary policies. A policy is said to be stationary if the action it chooses at time *n* only depends on the state of the process at time *n*, so stationary policy is a function $\pi : S \rightarrow A$. If policy π is employed, then $\{X_n, n = 0, 1, \ldots\}$ is a Markov chain with transition probabilities $P_{ij}(\pi(i))$, it is for this reason the process is called a Markov decision process. A stationary randomized policy can be considered as a map $\varphi : S \rightarrow \mathcal{P}(A)$ ($\mathcal{P}$ (...) = the space of probability vectors on "..."), which gives the conditional probabilities of a^j given X_n for all $1 \leq j \leq m$.

To determine policies that are in some sense optimal, we consider the infinite horizon discounted return as our optimal criterion. This criterion assumes a discount factor $\alpha, 0 < \alpha < 1$, and among all policies π, attempts to maximize V^π where

$$V^\pi(i) = E^\pi[\sum_{n=0}^{\infty} R(X_n, a_n)\alpha^n \mid X_0 = i], \quad i \in S \quad (2)$$

The function $V^\pi : X \rightarrow \mathcal{R}$ is called the value function for policy π. The use of a discount factor is economically motivated by the fact that the value of money earned tomorrow is worth discounted amount today. The optimal value of the value function, is:

$$V^*(i) = \max_\pi V^\pi(i), \quad i \in S \quad (3)$$

An important equation that V^* satisfies is Bellman's optimality equation[13]:

$$V(i) = \max_a [R(i,a) + \alpha \sum_j P_{ij}(a)V(j)], \quad i \in S \qquad (4)$$

The fact that V^* satisfies Bellman's equation can be explained as follows. In the above equation the term in square brackets on the right hand side is the total reward that one would get if action a is chosen at the first time step and then the system performs optimally in all future time steps. Clearly, this term cannot exceed $V^*(i)$ since that would violate the definition of V^*, thus V^* satisfies the Bellman's equation. It is also known that Bellman's equation has a unique solution [13]. Now the optimal decision problem turns out to be finding V^*.

2.2 Dynamic Programming Techniques

Two standard approaches to compute V^* are *value iteration* and *policy iteration*.

Value iteration: This starts with an initial guess V_0 for the optimal V^* and recursively iterates as per

$$V_{n+1}(i) = \max_a [R(i,a) + \alpha \sum_j P_{ij}(a)V_n(j)], \; i \in S \qquad (5)$$

for $n \geq 0$. Using Banach contraction mapping theorem, it is easy to show that $V_n \rightarrow V$ at an exponential rate.

Policy iteration: This starts with initial stationary (randomized) policy $\pi_0 : S \rightarrow A$, for an optimal policy, it does iteratively for $n \geq 0$ as follows:
Step 1: Given $\pi_n(.)$, find $V_n : S \rightarrow \mathcal{R}$, satisfying

$$V_n(i) = R(i, \pi_n(i)) + \alpha \sum_j P_{ij}(\pi_n(i))V_n(j), \; i \in S \qquad (6)$$

Step 2 : Find

$$\pi_{n+1}(i) \in argmax(R(i,.) + \alpha \sum_j P_{ij}(.)V_n(j)), \qquad i \in S \qquad (7)$$

then $V_n \rightarrow V$ in finitely many steps.

Convergence issues: In value iteration, we solve the nonlinear system of equations (5). Define the nonlinear value iteration operator, *B*, in vector-form as

$$B(V) = \max_\pi (R(\pi) + \alpha P(\pi)V), \; V \in \mathcal{R}^{|S|} \qquad (8)$$

Component wise, *B* can be written as follows:

$$B_i(V) = \max_a [R(i,a) + \alpha \sum_j P_{ij}(a)V(j)], i \in S. \qquad (9)$$

For $\alpha < 1$, *B* is a contraction operator because $\forall V \in \mathcal{R}^{|S|}, \|B(V) - V^*\| \leq \alpha \|V - V^*\|$. Therefore, the value iteration algorithm can be proven to converge to V^* by using Banach contraction mapping theorem, and $V_n \rightarrow V$ at an exponential rate.

In policy iteration, we evaluate a fixed stationary policy π, which requires solving a linear system of equations (6). Here also, we can define an operator B^π as

$$B^\pi(V) = R(\pi) + \alpha P(\pi)V \qquad (10)$$

For $\alpha < 1$, the operator B^π is a contraction operator because $\forall V \in \mathcal{R}^{|S|}, \|B^\pi(V) - V^\pi\| \leq \alpha \|V - V^\pi\|$. So V_π is the unique solution to $V = B_\pi(V)$. Of course, the operators B and B^π require knowledge of the state transition probabilities. For getting more insight on the above issues, refer [14], [2].

One has to have complete knowledge of transition probabilities $P_{ij}(a)$ in order to solve the MDP problem by applying the above techniques. When we do not know the environment dynamics (i.e. transition probabilities) we can use reinforcement learning algorithms like Q-learning [18], actor-critic algorithm [8], which are stochastic approximation counterparts of value iteration and policy iteration, respectively. These algorithms are used when the transition probabilities $P_{ij}(a)$ are not explicitly available but a transition with a prescribed probability can be *simulated*. We now discuss the intuition behind appropriate simulation-based versions of the above two algorithms, for more insights one can refer [14]. The first, of course, is that $P_{ij}(a)$ is replaced by a simulated transition as per the prescribed probabilities $P_{ij}(a)$. In order for this to work, the algorithms should do some averaging. This is ensured by using an incremental version which makes only a small change in current iterates at each step, weighted by a stochastic approximation-like decreasing step size. Q-learning is a simulation-based version of the value iteration, where one works not with value function V, but the Q-value defined by

$$Q(i,a) = R(i,a) + \alpha \sum_j P_{ij}(a)V(j), i \in S, a \in A. \qquad (11)$$

Thus Q satisfies

$$Q(i,a) = R(i,a) + \alpha \sum_j P_{ij}(a) \max_b Q(j,b) \qquad (12)$$

The iterative scheme is

$$Q_{n+1}(i,a) = R(i,a) + \alpha \sum_j P_{ij}(a) \max_b Q_n(j,b) \qquad (13)$$

If $P_{ij}(a)$ are not available, in Q-learning one replaces the above equation by

$$\begin{aligned} Q_{n+1}(i,a) &= (1-\gamma_n)Q_n(i,a) + \gamma_n(R(i,a,\xi_n(i,a)) \\ &\quad + \alpha \max_b Q_n(\xi_n(i,a),b) \end{aligned} \qquad (14)$$

We choose a sequence of step sizes γ_n such that $\sum \gamma_n = \infty$ and $\sum \gamma_n^2 < \infty$ and $\xi_n(i,a)$ is a random variable with law $P_{i.}(a), \gamma_n$ is the step size at the n^{th} iteration. We start the algorithm by initializing the Q function and by simulation we get $\xi_n(i,a)$ and we calculate $R(i,a,\xi_n(i,a))$, iteratively we update the Q function. Observe that the Q-learning algorithm demands a reinforcement signal, that is $R(i,a,\xi_n(i,a))$ for learning the Q function, refer [18] for more details of convergence issues of the Q-learning algorithm. Actor-critic type algorithms are simulation-based version of the policy iteration. The linear system of equations in *step 1* is replaced by an iterative scheme for its solution

$$V_n^{m+1} = R(i,\pi_n(i)) + \alpha \sum_j P_{ij}(\pi_n^m(i)) V_n^m(i), i \in S, m \geq 0 \quad (15)$$

where m is being updated for each fixed n, the value V_n (which is $V_n^m \rightarrow V_n$) is then passed to *step2* for updating the policy. The crux of the algorithm proposed in [8] is to achieve this two-tier structure by using two different time scales is as follows. We operate with stationary randomized policies rather than stationary policies. Let $\{a(n)\}, \{b(n)\}$ be decreasing sequences in (0,1) satisfying $\sum_n a(n) = \sum_n b(n) = \infty, \sum_n a(n)^2, \sum_n b(n)^2 < \infty, a(n) = o(b(n))$. Fix $a_0 \in A$ and let Γ denote the projection of $\overline{a}_n$, $(r-1)$-vector $([\pi(i,a^1),\ldots,\pi(i,a^r)])$ onto the simplex $D = \{[x_1,\ldots,x_{r-1}] | x_i \geq 0, \forall i, \sum_i x_i \leq 1\}$. The algorithm starts with an initial pair $V_0(.)$ and $\pi_0(i,a)$, $i \in S, a \in A \backslash \{a_0\}$, and iterates according to

$$V_{n+1}(i) = (1-b(n))V_n(i) + b(n) \sum_a [R(i,a,\xi_n(i,a)) + \alpha V_n(\xi_n(i,a))] \pi_n(i,a) \quad (16)$$

$$\pi_{n+1}(i,a) = \pi_n(i,a) + a(n) \sum_a [\, R(i,a,\xi_n(i,a)) + \alpha V_n(\xi_n(i,a)) \pi_n(i,a) - V_n(i)] \pi_n(i,a) \quad (17)$$

Evaluating the value function for a particular policy is considered as criticism and updating the present policy is considered as action, that is why it is called as actor-critic algorithm, refer [8] for convergence issues of the actor-critic algorithm.

3 Single Seller Model

Figure 1 depicts the single-seller case. Here, we have a seller who wishes to maximize his revenue by using dynamic pricing methodology. We assume the following:

- The seller maintains a finite inventory of the product. $I_{\max}$ is the maximum inventory. The seller follows a naive, fixed reorder policy whereby, each time the inventory level drops to a level r, he would order a replenishment of size $I_{\max} - r$. The replenishment lead time (time elapsed between placement of replenishment order and the arrival of the items) exponentially distributed.

- The seller uses both price-dispersion by changing the price (p_1) of a unit of the product dynamically. Also the seller uses price discrimination through volume discounts. For the purposes of this paper, we assume volume discounts of the type: buy-two-get-three (pay $2p_1/3$ for 3 units of the product). More general varieties of volume discounts can also be modeled.

- The buyers are distinguished into two categories, following Varian [17]: captives and shoppers. Captives are mature buyers whose buying patterns are not influenced by availability or otherwise of volume discounts. We assume they always buy one unit of the product. On the other hand, shoppers are influenced by and go after volume discounts. We assume q percent of the buyers are captives and the rest are shoppers. Captives wait in Queue1 and shoppers wait in Queue2.

- Captives are given higher priority by the sellers in the following sense: if the item is not available in inventory when a captive arrives, the seller would provide the incoming captive with a price quote and lead time quote. If the quoted lead time and price are within his interest, he will commit to purchasing the item (if the lead time > 0 then he has to wait in Queue1), otherwise he leaves the system. An incoming shopper does not get a quote regarding lead time or price. A shopper may choose to wait in Queue2 if he likes the present price p_2 and he can balk from Queue2 according to an exponentially distributed waiting time if he does not get service within that time. If a shopper is offered the item then he has to pay that price p_2 (if shopper does not like the price then also he can balk from the system).

Let us consider the revenue optimization problem of the seller in the above model. To describe the state of the system, we use queue lengths and the inventory level at the seller $(q_1(t), q_2(t), I(t))$. The possible set of actions at any state is the set A, from which seller can choose a price (p_1) to display. The prices will be changed whenever a customer enters (or leaves) and whenever a replenishment arrives at the seller inventory. Let $p_{ij}(a)$ be the transition probability of the system from state i to state j when pricing action a at state i, and transition time from i to j be according to a distribution $F_{ij}(a)$. The rest of the sellers in the market and

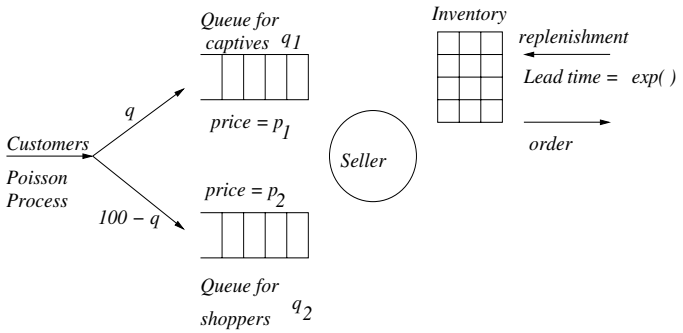

Figure 1. Single-seller Model

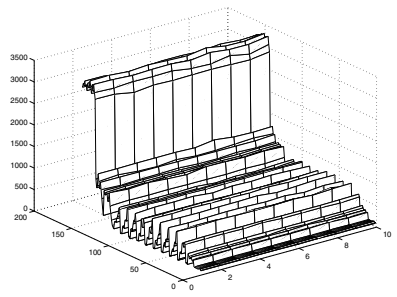

Figure 2. Q-Function

the distributor have indirect influence on these distributions and cannot be expressed explicitly. Let $r(i, a, j)$ denote the reward to the seller for pricing action a at state i and if it results state j. The value function (discounted expected reward) for any stationary policy $\pi : S \rightarrow A$:

$$V^{\pi}(i) = [r(i, \pi(i)) + \sum_{j} p_{ij}(\pi(i)\alpha V^{\pi}(j)] \quad (18)$$

Where $r(i, \pi(i))$ is the expected immediate reward at state i and policy π. Where α (which takes values from set [0,1]) is the discount factor on future cumulative reward $V^{\pi}(j)$ if system transfers from i to j. We assume that α's value depends on i, j and time (t) taken for the system to move from state i to state j. So the expected value of the random variable $\alpha V^{\pi}(j)$ can be written as

$$E[\alpha V^{\pi}(j)] = \int_{0}^{\infty} e^{-\beta t} V^{\pi}(j) dF_{ij}(t \mid \pi(i)) \quad (19)$$

$$\begin{aligned} V^{\pi}(i) &= [r(i, \pi(i)) + \sum_{j} p_{ij} \\ &\int_{0}^{\infty} e^{-\beta t} V^{\pi}(j) dF_{ij}(t \mid \pi(i))] \end{aligned} \quad (20)$$

Bellman's optimality condition can be written as

$$\begin{aligned} V^{*}(i) &= \max_{\pi(i)} [r(i, \pi(i)) + \sum_{j} p_{ij} \\ &\int_{0}^{\infty} e^{-\beta t} V^{*}(j) dF_{ij}(t \mid V^{*}(i))] \end{aligned} \quad (21)$$

Let us consider the learning problem of the seller in the above model. We assume that the seller does not have any information about the strategies of other sellers in the retail market and preferences of the buyers who are approaching the retail market. We use stochastic approximation counterpart of the value iteration algorithm, that is Q-learning algorithm.

$$\begin{aligned} Q_{n+1}(i, a) &= Q_n(i, a) + \gamma_n [r(i, a, j) + \\ &e^{-\beta T} \max_{b} Q_n(j, b) - Q_n(i, a)] \end{aligned} \quad (22)$$

Where T is the sample time taken by the system for moving from state i to state j while collecting samples by doing simulation. If seller takes action a at state i and system moves to state j then $r(i, a, j)$ is the reward (amount of business done + inventory holding cost) to the seller, while collecting samples by doing simulation.

3.1 Simulation Experiment for Single Seller Market

We use Q-learning algorithm [18] for learning the best strategy at every state of the system. We simulate and study the above model as shown in Figure 1, by considering action (price) set A = {9, 9.5, 10, 10.5, 11, 11.5, 12, 12.2, 12.5, 13}, maximum queue lengths are assumed to be 10 for both Queue1 and Queue2. The maximum inventory level is assumed to be 20 with fixed reorder (of size 10) point at $r = 10$. There are 161 states in the state space of this process. We assume that $q = 0.2$, that is, 20 percent of the incoming consumers are captives. We consider consumers arriving in Poisson fashion with mean inter-arrival time 30 minutes, and zero service time at the seller. Private values of upper limits on price and waiting time for an incoming consumer are uniformly distributed over (8, 14], (0, 12 hours] respectively. We consider exponential replenishment lead time for reorders with a mean of 10 hours. The frustrated shopper drops out of the system with an exponential waiting time having a mean of 5 hours. Q-function values are shown in Figure 2, where Q values are plotted against (state, action) pairs. By knowing the Q-function seller can easily compute the best possible prices for a given situation (state). We use ϵ greedy policy [15] while using the Q-learning algorithm, with discount factor β set at 0.001.

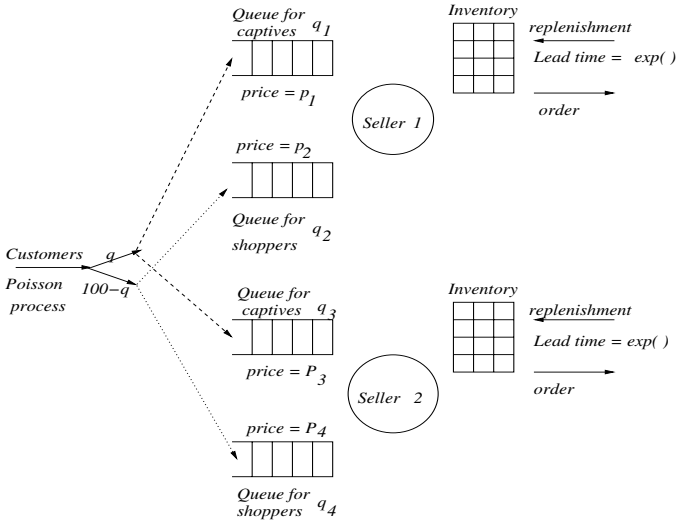

Figure 3. Two-seller Model

4 Two Seller Model

Figure 3 depicts the two-seller case. Here, two competing sellers wish to maximize their respective revenues by using RL-based adaptive behavior. All assumptions about individual sellers and buyers are the same as explained in the single seller model. We further assume that every captive is associated with a utility function that combines price and delay in a form:

$$Utility = [(1-q)(p_b-p)+q(w_b-w)]\Theta(p_b-p)\Theta(w_b-w) \quad (23)$$

where $\Theta(x) = 1$ if $x > 0$, $\Theta(x) = 0$ otherwise and $0 \leq \alpha \leq 1$. Each buyer has his own upper-limits, p_b, w_b, on price and waiting time respectively and are assumed to be *i.i.d* and uniformly distributed over intervals $(0, p_{max}]$ and $(0, w_{max}]$. A captive buys from a seller where his utility is more and positive. If the captive cannot find positive utility then, he drops from the system. Since an incoming shopper does not get a quotation regarding lead time or price, the shopper observes the prices at both the sellers and joins in a queue (Queue2 or Queue4), where the price is less and with in his price limits and he can balk from those queues according to some waiting time distribution (exponentially distributed), if he does not get service with in that time. If a shopper is offered the item he has to pay that time price p_2 or p_4. If the shopper does not like the price then he can leave the system or balk to the other shop if gets the item immediately and at a lower price. We assume that each seller is equipped with a mechanism to observe queue status and inventory at the other seller. Since the system is simultaneously controlled by more than one decision maker, we model the above problem as a *stochastic game* with state description $(q_1, q_2, q_3, q_4, I_1, I_2)$. Sellers simultaneously choose actions from the grid $A \times A$. The possible set of actions at any state is the set $A = \{a^1, ..., a^m\}$, from which the first seller chooses prices (p_1 and $p_2 = 2(p_1)/3$) and the second seller chooses prices (p_3 and $p_4 = 2(p_3)/3$). The prices will be changed whenever a customer enters (or leaves) the system and on arrival of an inventory lot to one of the seller's shop from the distributor. Let $p_{ij}(a)$ be the transition probability of the system from state i to state j when pricing action a at state i is chosen. $R^l[i,[a_1,a_2]]$ is player l reward if both the players choose action vector $[a_1, a_2] \in A \times A$ at state i and the system moves to the state j. Since our aim aim is to consider the game when transition structure is not known, and employ simulation-based approximation methods based on reinforcement learning, we need the following formal development of a Markovian game. Let $X(t)$ be the Markovian game and $\tau_k, k = 0, 1, ...$ be the sequence of event epochs, with τ_n denoting an instance when either a departure or arrival of a buyer at any of the two sellers or arrival of inventory at any of the two sellers happens. We use X_k to denote X_{τ_k} and a_k to denote a_{τ_k}. Let $\pi = [\pi^1(.,.), \pi^2(.,.)] \in (\mathcal{P}(A))^{2m}$ be any stationary randomized strategy pair of the players. $\mathcal{P}(A)$ denotes the space of distributions over A. The policy evaluation function of player $l, l = 1, 2$ is as follows.

$$\begin{aligned} V_\pi^l(i) &= E_\pi[\sum_{n=0}^{\infty} e^{-\beta(\tau_1+...+\tau_{n-1})} \\ &\quad R^l(X(\tau_{n-}, Z_n, X(\tau_n)) \mid X_1 = i], \forall i \quad (24) \end{aligned}$$

where Z_n is the control process at event epoch τ_n with law as prescribed by the stationary randomized policy π. We call the policy $\pi(.,.)$ a *Nash equilibrium* if for every $l, V_\pi^l(i) \leq V_{\overline{\pi}^k(i)}^k \forall i$ whenever, $\overline{\pi}^k(.,.) = \pi^l(.,.)$ for $l \neq k$. It is known that a Nash equilibrium in this sense exists [4]. If we freeze the policies of one seller then it becomes a Markov decision process for the other seller. These facts motivate us to use actor-critic type of learning paradigm for stochastic games to learn such strategies. Also, in the case where both the sellers try to learn their Nash equilibrium strategies following best response dynamics, it can be hoped that both will converge to a Nash equilibrium if seller 1 observes seller 2 as quasi-static and seller 2 observes seller 1 as playing equilibrium strategy in their pursuit for mutual best responses. With this intuition, we devise two similar actor-critic (one actor and one critic for each seller) learners that operate on different time scales for updates.

$$\begin{aligned} V_{n+1}^l(i) &= (1 - b^l(n))V_n^l(i) + b^l(n) \sum_a [R^l(i, a, \\ &\quad \xi_n(i, a)) + e^{-\beta T} V_n^l(\xi_n(i, a))]\pi_n^l(i, a) \quad (25) \end{aligned}$$

$$\pi_{n+1}^l(i, a) = \pi_n^l(i, a) + a^l(n) \sum_a [R^l(i, a, \xi_n(i, a))$$

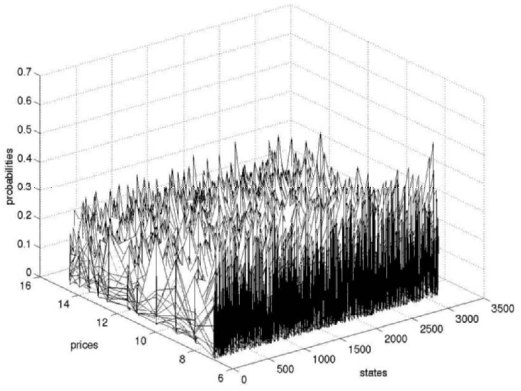

Figure 4. Seller 1 Policy Function

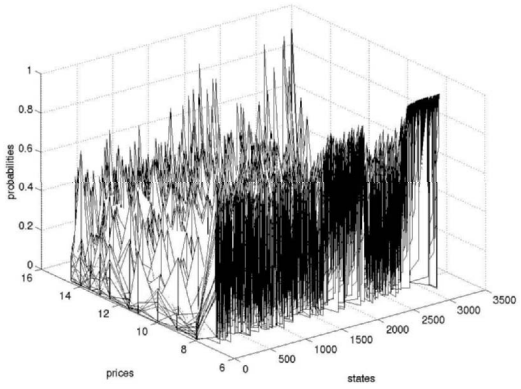

Figure 5. Seller 2 Policy Function

$$+e^{-\beta T}V_n^l(\xi_n(i,a))\pi_n^l(i,a) - V_n^l(i)]\pi_n^l(i,a) \quad (26)$$

In addition to the standard condition $a^l(n) = o(b^l(n)), l = 1, 2$ and $a^2(n) = o(a^1(n))$ for making seller 2 as a slower learner. The assumption that one seller is a faster learner than the other is a valid model when the competing sellers differ in their information acquisition capabilities.

4.1 Simulation Experiment for Two Seller Market

We assume the action (price) set A = {7, 8, 9, 10, 11, 11.5, 12.5, 13, 13.5, 14, 14.5}, maximum queue lengths are 5 at each queue, and maximum inventory is 10 at each seller, with fixed reorder (of size 5) point at $r = 5$ at each seller. Rest of the system parameters are identical to that of the previous model. We use two actor-critic learners with learning rate parameters $a^1(n) = 1/n, a^2(n) = 1/n^{1.5}, b^1(n) = 1/n^{0.6}$ and $b^2(n) = 1/(n^{0.6} + 10)$. The discount factor β is set at 0.001. The system (no. of states are 3137) is simulated over 12000 iterations. Figure 4 and Figure 5 show the best randomized strategies against (state, action) pairs, for seller-1 and seller-2 respectively. Table 1 shows the convergence strategies for two randomly picked states (5,5,0,0,0,10) and (0,0,5,5,10,0) of both the sellers. It is interesting to note that at state (5,5,0,0,0,10), seller-1 randomizes his prices in high price domain and seller-2 randomizes his prices in low price domain. This result can be explained as follows: since the shop of seller-1 is overcrowded, he tries to discourage the incoming buyers by displaying a high price. On the other hand seller-2 does not have customers, so for reducing the inventory cost he displays a low price for attracting more incoming buyers. A reverse trend can be observed at the state (0,0,5,5,10,0).

5 Conclusions and Future Work

In this paper, we have shown how reinforcement learning based techniques can be used in solving the dynamic pricing problem in retail markets. For the single-seller problem, we used the Q-learning algorithm and for two-seller problem we used two actor-critic learners. The models can be generalized to the case of more than two sellers. We believe this is a promising approach to solving the dynamic pricing problem in retail market environments where typically the information available to the agents is very limited.

There are several directions for future work. First of all, some of the assumptions made by us in the retail market model need to be relaxed: (1) nature of volume discounts (2) nature of inventory policy (3) assumptions regarding shoppers and captives. Secondly, safety stock to be maintained by sellers can be introduced as a decision variable and the model will then become much more interesting and complex. Convergence of the learning algorithms used is another important area of investigation which has engaged researchers in machine learning for quite sometime now.

References

[1] M. Bichler, R.D. Lawrence, J. Kalagnanam, H.S. Lee, K. Katircioglu, G.Y. Lin, A.J. King, and Y.Lu. Applications of flexible pricing in business-to-business electronic commerce. *IBM Systems Journal*, Volume 41, Number 2, 2002, 287-302.

[2] D. P. Bertsekas and J. Tsitsiklis. *Neuro-dynamic Programming*, Athena Scientific, September 1996.

[3] V. S. Borkar. Reinforcement learning in Markovian evolutionary games, *Advances in Complex Systems*, Volume 5, pp. 55-72, 2002.

Price	Policy of Seller 1 for state (5,5,0,0,0,10)	Policy of Seller 2 for state (5,5,0,0,0,10)	Policy of Seller 1 for state (0,0,5,5,10,0)	Policy of Seller 2 for state (0,0,5,5,10,0)
7.000000	0.0414408632	0.9956925511	0.4387517571	0.0448260903
8.000000	0.0298048202	0.0004108555	0.1043105870	0.0557303056
9.000000	0.0404702425	0.0003891649	0.0327194482	0.0660034865
10.000000	0.0500240512	0.0004402130	0.0656689331	0.1487187743
11.000000	0.0500319116	0.0003882047	0.0480695516	0.0244497284
11.500000	0.0585561171	0.0004617955	0.0483151674	0.0185835753
12.500000	0.0885034800	0.0004339077	0.0311893784	0.0777120143
13.000000	0.0999099761	0.0004741517	0.0475795642	0.0741290823
13.500000	0.1136456281	0.0004180684	0.0897262841	0.0099835452
14.000000	0.1139652133	0.0004521427	0.0616267025	0.0032900197
14.500000	0.3136477470	0.000438932	0.0320425406	0.4765734076

Table 1. Policies of sellers at some individual states

[4] A. Federgruen. On N-person stochastic games with denumerable state space, *Advances in Applied Probability*, Volume 10, pp. 452-471, 1978.

[5] M. Gupta, K. Ravikumar, and M. Kumar. Adaptive strategies for price markdown in a multi-unit descending price auction: A comparative study. *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pp. 373-378, 2002.

[6] J. C. Hu and M. Wellman. Multi-agent reinforcement learning: Theoretical framework and an algorithm, *Proceedings of the Fifteenth International Conference on Machine Learning*, Morgan-Kaufmann, pp 242-250, 1998.

[7] J. O. Kephart and A. R. Greenwald. Shopbot Economics, *Proceedings of the Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 208-220, 1999.

[8] V. R. Konda and V. S. Borkar. Actor-critic type learning algorithms for Markov decision processes, *SIAM Journal on Control and Optimization*, Volume 38, pp. 94-123, 1999.

[9] R.D. Lawrence. A machine-learning approach to optimal bid pricing. IBM Research Report, 2002.

[10] M. L. Littman. Markov Games as a Framework for Multi-agent Reinforcement Learning, *Proceedings of the Eleventh International Conference on Machine Learning*, Morgan-Kaufmann, pp 157-163, 1994

[11] M. L. Littman. Friend-or-foe Q-learning in general-sum games, *Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan-Kaufmann, pp. 322-328, 2001.

[12] K. Ravikumar, G. Batra, and R. Saluja. Multi-Agent Learning for Dynamic Pricing Games of Service Markets, Communicated.

[13] S. M. Ross. *Introduction to Stochastic Dynamic Programming*, Academic Press, 1983.

[14] S. Singh. Learning to solve Markovian Decision Processes, Ph.D Dissertation, University of Michigan, Ann Arbor, 1994.

[15] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[16] G. Tesauro and J. O. Kephart. Pricing in agent economies using multi-agent Q-learning, *Proceedings of Workshop on Decision Theoretic and Game Theoretic Agents*, London, England, July 1999.

[17] H. R. Varian. A Model of Sales, *The American Economic Review*, pp. 651-59, September 1980.

[18] C. J. C. H. Watkins and P. Dayan. Q-learning, *Machine Learning*, 8, 279-292, 1992.