강화 학습에서의 탐색과 이용의 균형을 통한 범용적 온라인 Q-학습이 적용된 에이전트의 구현

(Implementation of the Agent using Universal On-line Q-Learning by Balancing Exploration and Exploitation in Reinforcement Learning)

박 찬 건 † 양 성 봉 ††
(Chan-Geon Park) (Sung-Bong Yang)

요 약 shopbot이란 온라인상의 판매자로부터 상품에 대한 가격과 품질에 관한 정보를 자동적으로 수집함으로써 소비자의 만족을 최대화하는 소프트웨어 에이전트이다. 이러한 shopbot에 대응해서 인터넷상의 판매자들은 그들에게 최대의 이익을 가져다 줄 수 있는 에이전트인 pricebot을 필요로 할 것이다.

본 논문에서는 pricebot의 가격결정 알고리즘으로 비 모델 강화 학습(model-free reinforcement learning) 방법중의 하나인 Q-학습(Q-learning)을 사용한다. Q-학습된 에이전트는 근시안적인 최적(myopically optimal 또는 myoptimal) 가격 결정 전략을 사용하는 에이전트에 비해 이익을 증가시키고 주기적 가격 전쟁(cyclic price war)을 감소시킬 수 있다. Q-학습 과정 중 Q-학습의 수렴을 위해 일련의 상태-행동(state-action)을 선택하는 것이 필요하다. 이러한 선택을 위해 균일 임의 선택방법 (Uniform Random Selection, URS)이 사용될 경우 최적 값의 수렴을 위해서 Q-테이블을 접근하는 회수가 크게 증가한다. 따라서 URS는 실 세계 환경에서의 범용적인 온라인 학습에는 부적절하다. 이와 같은 현상은 URS가 최적의 정책에 대한 이용(exploitation)의 불확실성을 반영하기 때문에 발생하게 된다.

이에 본 논문에서는 보조 마르코프 프로세스(auxiliary Markov process)와 원형 마르코프 프로세스(original Markov process)로 구성되는 혼합 비정적 정책 (Mixed Nonstationary Policy, MNP)을 제안한다. MNP가 적용된 Q-학습 에이전트는 original controlled process의 실행 시에 Q-학습에 의해 결정되는 stationary greedy 정책을 사용하여 학습함으로써 auxiliary Markov process와 original controlled process에 의해 평가 측정된 최적 정책에 대해 1의 확률로 exploitation이 이루어질 수 있도록 하여, URS에서 발생하는 최적 정책을 위한 exploitation의 불확실성의 문제를 해결하게 된다. 다양한 실험 결과 본 논문에서 제한한 방식이 URS 보다 평균적으로 약 2.6배 빠르게 최적 Q-값에 수렴하여 MNP가 적용된 Q-학습 에이전트가 범용적인 온라인 Q-학습이 가능함을 보였다.

키워드 : 강화 학습, Q-학습, 다중 에이전트 시스템, 에이전트 경제, 샵봇, 프라이스봇

Abstract A shopbot is a software agent whose goal is to maximize buyer's satisfaction through automatically gathering the price and quality information of goods as well as the services from on-line sellers. In the response to shopbots' activities, sellers on the Internet need the agents called pricebots that can help them maximize their own profits.

In this paper we adopts Q-learning, one of the model-free reinforcement learning methods as a price-setting algorithm of pricebots. A Q-learned agent increases profitability and eliminates the cyclic price wars when compared with the agents using the myoptimal (myopically optimal) pricing strategy. Q-learning needs to select a sequence of state-action pairs for the convergence of Q-learning. When the uniform random method in selecting state-action pairs is used, the number of accesses to the

† 학생회원 : 연세대학교 컴퓨터과학과
cgpark@cs.yonsei.ac.kr
†† 비 회 원 : 연세대학교 컴퓨터산업공학부 교수
yang@cs.yonsei.ac.kr
논문접수 : 2002년 3월 22일
심사완료 : 2003년 3월 11일

Q-tables to obtain the optimal Q-values is quite large. Therefore, it is not appropriate for universal on-line learning in a real world environment. This phenomenon occurs because the uniform random selection reflects the uncertainty of exploitation for the optimal policy.

In this paper, we propose a Mixed Nonstationary Policy (MNP), which consists of both the auxiliary Markov process and the original Markov process. MNP tries to keep balance of exploration and exploitation in reinforcement learning. Our experiment results show that the Q-learning agent using MNP converges to the optimal Q-values about 2.6 times faster than the uniform random selection on the average.

Key words : Reinforcement learning, Q-learning, Adaptive multi-agent systems, Agent economies, Shopbot, Pricebot

1. 서 론

오늘날 인터넷 상에서 소비자 상품과 서비스가 급속하게 증가되고 복잡해지면서 소비자들은 신속하고 자동적으로 유용한 정보를 찾아줌으로써 편리성과 효율성을 제공하는 소프트웨어 에이전트에 점차 의존하고 있다. shopbot은 이와 같이 온라인상의 판매자들을 질의해서 인터넷상의 상품과 서비스의 가격이나 다른 속성들을 자동적으로 수집하는 웹 에이전트이다. 현재 shopbot이 다루는 상품 종류의 범위가 크게 증가하고 있으며 표 1에서는 몇몇 shopbot에서 다루고 있는 상품 범위 및 그 특징을 간략하게 보여준다[1].

shopbot이 소비자의 이익과 만족을 위한 에이전트라고 한다면, 판매자의 이익을 위해 자동적인 가격 결정 전략을 사용하는 소프트웨어 에이전트인 *pricebot*이 있다. 현재 인터넷 상의 pricebot들 중 대표적으로 도서를 판매하는 Book.com이 있다. Book.com은 Amazon.com, Borders.com과 Barnesandnoble.com에 질의를 하고 이들 사이트에서 제시하는 가격을 비교해서 세 개의 질의된 가격 중 가장 낮은 가격에서 소매가격의 1%를 낮춘 가격을 제시한다.

유익한 가격 결정을 하기 위해서는 적절한 가격 결정 정책이 반드시 필요하다. 본 논문에서 기술될 에이전트 경제는 인터넷을 기반으로 이루어지는 것을 가정한다. 인터넷 상의 에이전트들은 서로에 대해 잘 알지 못하며 또한 학습하는 동안 환경이 지속적으로 변할 수 있다. 따라서 on-line 학습을 위한 비 모델 학습방법이 필요하다.

본 논문에서는 pricebot이 다른 에이전트와의 경쟁에서 최적의 가격 결정 정책을 학습하도록 하기 위한 가격 결정 알고리즘으로 Q-학습(Q-learning) [2]을 채택한다. Q-학습은 환경 내에서 감지하고 행동하는 자치적 에이전트가 목적을 달성하기 위한 최적의 행동(action)을 선택하기 위한 비 모델 강화 학습(reinforcement learning) 방법 중의 하나이다. 마르코프 결정 프로세스(Markov decision process)에 대해서 Q-학습 에이전트는 보상 함수(reward function) 또는 상태 변화 함수(state transition function)를 모르더라도 직접적으로 환경과의 교류를 통해서 최상의 정책을 학습할 수 있다.

Q-학습된 에이전트는 장기적인 행동들에 대한 결과를 예상할 수 있는 능력을 가지게 됨으로써, Q-학습된 에이전트는 예측 능력을 가지지 않은 근시안적인 최적(myopically optimal 또는 myoptimal) 가격 결정 전략이 에이전트에 사용됐을 때 발견되는 끝이 없이 주기적으로 반복되는 가격 전쟁(price war)의 크기(amplitude)를 줄이고 더 많은 이익을 얻을 수 있게 한다[3].

다른 종류의 학습에서는 일어나지 않지만 강화 학습에서 발생하는 문제들 중 하나는 탐색(*exploration*)과 이용(exploitation) 사이의 균형 문제이다. 에이전트는 보상을 얻기 위해 이미 알고 있는 것을 exploitation해야 하지만 역시 미래에 더 좋은 행동 선택을 위해서

표 1 shopbot 관련 상품의 범위 및 특징

shopbot	상품	특징
Shopper.com	컴퓨터 관련 제품	1,000,000개의 가격 비교
Dealpilot.com (과거Cses.com)	책, CD, 영화	가격과 배달시간의 수집, 대조, 및 정렬
Mysimon.com	전자제품, 의복, 장난감, 잡화, 학용품	다중 속성에 대한 질의를 사용하며 유틸리티에 따라 상품 정렬

exploration도 해야 한다. 그러나 어떤 목적을 달성하기 위해서는 exploration과 exploitation 모두 배타적으로 이루어 질 수 없다. 어떤 하나의 행동을 선택할 때 exploration과 exploitation이 동시에 이루어지도록 하는 것은 불가능하기 때문에 충돌이 발생하게 되며, exploration하는 것이 바람직한지 exploitation을 하는 것이 더 바람직한지는 평가된 값의 정확성, 불확실성 등과 같은 복잡한 요인에 의해 판단 될 수 있다. 강화 학습에서 exploration과 exploitation의 균형은 다른 학습 방법과 구별되는 중요한 문제이다.

Q-학습 에이전트는 최적의 정책 결정을 위해 일련의 상태-행동을 선택한다. 이때 에이전트가 Q-학습의 수렴을 위해 균일 임의 선택방법 (Uniform Random Selection, URS) [3, 8, 9]을 사용하게 되면, 최적의 정책에 대한 exploitation의 불확실성을 반영함으로 Q-테이블을 접근하는 회수가 증가하게 된다. 따라서 최적 정책 결정을 위한 URS는 강화 학습에서의 exploration과 exploitation의 균형문제를 해결 할 수 있는 적절한 방법이 아니며, 비록 Q-학습이 온라인 학습에 사용될 수 있는 방법이긴 하나 상태 접근 회수의 증가로 실 세계 환경에서 적용되기 어렵다.

본 논문에서는 이와 같이 강화 학습에서 exploration과 exploitation사이에서의 균형이 이루어지도록 하기 위해 혼합 비정적 정책 (Mixed Nonstationary Policy, MNP)을 제안한다. Q-학습 에이전트에 MNP를 적용하면 original controlled process의 실행 시간 동안 Q-학습에 의해 결정되는 stationary greedy 정책을 사용하여 학습함으로써 auxiliary Markov process와 original controlled process에 의해 평가 측정된 최적 정책에 대해 1의 확률로 exploitation이 이루어질 수 있다. 다양한 실험 결과 MNP가 URS 보다 약 3.47배 빠르게 최적 정책에 수렴하였으며, 이는 MNP를 적용한 Q-학습 에이전트가 범용적인 온라인 학습이 가능하다는 것을 보여준다.

이 후의 본 논문은 다음과 같이 구성되어 있다. 2장에서는 shopbot과 pricebot으로 구성되는 경제모델의 구조에 대해 소개하고 pricebot들이 myoptimal 가격 결정 전략을 사용했을 때의 가격 결정 변화 추이를 보인다. 3장에서는 단일 에이전트 Q-학습 알고리즘을 설계하는데 필요한 사항들에 대해 기술하고 Q-학습된 에이전트와 myoptimal 에이전트와의 경쟁 환경 속에서 어떤 결과를 보이는지를 가격 결정 변화와 평균 이익의 측면에서 기술한다. 4장에서는 MNP를 제안, 설명하며 5장에서는 실험을 통해서, 제안된 방법을 사용한 Q-학습의 결과를 최적 값의 수렴을 위한 특정 상태 Q-테이블의 접근 회수의 관점에서 URS를 사용한 Q-학습과 비교하여 결과 및 성능 분석을 하며, 마지막으로 6장에서는 결론과 향후 관련 연구에 대해 설명한다.

2. SHOPBOT과 PRICEBOT 에이전트 경제 모델

2.1 모델

본 논문에서는 시뮬레이션을 위한 경제 모델을 위해 소비자들은 모든 온라인 판매자들의 특정 상품에 대한 가격과 다른 속성들을 자동적으로 수집하는 shopbot을 사용할 수 있고, 판매자들은 능동적인 가격 결정 알고리즘을 사용해서 최대의 이익을 가져다 줄 수 있는 pricebot을 사용하는 에이전트 경제를 채택한다. 이러한 경제 시장에는 S명의 판매자들이 판매를 위해 동질의 상품을 제공하고, 그 상품에 대해 관심이 있는 B명의 소비자가 존재하며 $B \gg S$ 임을 가정한다. 각각의 판매자 s는 p_s의 비율을 가지고 임의적인 시간에 가격 p_s를 재 조정 하는 반면, 각각의 소비자 b는 ρ_b의 비율로 임의 시간에 구입 주문을 하게 된다. 소비자 b의 상품 평가 값은 v_b이고, 판매자 s에 대한 생산 비용은 c_s이다. 해당 상품에 대한 소비자 b의 유틸리티는 다음과 같다.

$$u_b(p_s)=\begin{cases} v_b - p_s & \text{if } p_s \le v_b \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

우리는 소비자가 그들의 유틸리티만을 최대로 하려는 성향을 가지지 않는다고 가정한다. 그 대신에 소비자들은 판매자들을 임으로 선택해서 그들이 제시한 가격이 소비자의 상품 평가 값보다 낮으면 그 상품을 구입하는 '*Any Seller*' 전략과, shopbot을 사용해서 모든 판매자 중 가장 낮은 가격을 제시하는 판매자를 결정해서 그 최저 가격이 소비자의 상품 평가 값보다 낮으면 그 판매자를 거래를 위한 상대로 선택하는 '*Bargain Hunter*' 전략 중 한가지를 사용한다[4]. 이와 같이 소비자들의 전체 분포는 '*Any Seller*' 전략을 사용하는 소비자의 분포 w_x와 '*Bargain Hunter*' 전략을 사용하는 소비자들의 분포 w_y로 이루어지며 $w_x + w_y = 1$이 된다.

단위 시간 당 판매자 s의 기대 이익 π_s는 가격 벡터 p의 함수로 다음과 같다.

$$\pi_s(p)=(p_s - c_s)D_s(p) \quad (2)$$

단, $D_s(p)$는 판매자 s가 제공한 상품에 대한 요구 비율을 의미하며 따라서 다음 식과 같이 정의 할 수 있다.

$$D_s(p)=\rho B h_s(p) g(p_s) \quad (3)$$

이때, ρB는 전체 소비자의 요구 비율로서 $\rho = \sum_b \rho_b$ 이며, $h_s(p)$는 판매자 s가 소비자로부터 선택될 확률이고, $g(p_s)$는 $v_b \geq p_s$를 만족하는 소비자들의 비율이다. 일반성을 위해 $\rho B = 1$이라고 하면, 판매자 s의 기대 이익 π_s는 다음과 같이 나타낼 수 있다.

$$\pi_s(p) = (p_s - c_s) h_s(p) g(p_s) \quad (4)$$

$h_s(p)$는 소비자들이 그들의 잠재적 판매자로써 s를 선택할 확률이므로 다음과 같이 소비자 분포 (w_x, w_y)에 의해 좌우된다.

$$h_s(p) = w_x f_s^x(p) + w_y f_s^y(p) \quad (5)$$

$f_s^x(p)$는 *Any Seller* 전략을 사용하는 소비자들이 판매자 s를 선택할 확률을 의미하며, 이는 판매자들의 가격 순서에 독립적이므로 $1/S$ 이 된다. 그러나 *Bargain Hunter* 전략을 사용하는 소비자들이 s를 잠재적인 판매자로 선택할 확률 $f_s^y(p)$는 판매자들이 제시한 가격의 상대적인 순서에 의존적이므로 다음과 같이 정의될 수 있다.

$$f_s^y(p) = \frac{1}{\tau_s(p)+1} \delta_{\lambda_s(p),0} \quad (6)$$

단, $\tau_s(p)$는 s자신을 제외한 같은 가격을 제시하는 판매자들의 수이며, δ_{ij}는 Kronecker delta 함수로, $i = j$ 이면 1이고, 그렇지 않으면 0이고, $\lambda_s(p)$는 판매자 s보다 더 낮은 가격을 제시하는 판매자들의 수이다. 소비자의 평가 값과 관련해서 모든 소비자는 상품에 대해서 동일한 평가($v_b = v$)를 한다고 하면 $g(p) = \Theta(v - p)$ 라고 할 수 있다.

$$\Theta(v-p) = \begin{cases} 1 & \text{if } p \leq v \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

위에서 정의된 식들을 사용하여 판매자의 기대 이익 함수 π_s를 소비자가 사용하는 전략의 분포와 평가 값에 의해서 표현 할 수 있다. 모든 소비자 b에 대해 $v_b = v$ 이고, 모든 판매자 s에 대해 $c_s = c$라고 하면 다음과 같이 정의 된다.

$$\pi_s(p) = \begin{cases} (p_s - c) h_s(p) & \text{if } p_s \leq v \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$h_s(p) = w_x \frac{1}{S} + w_y \frac{1}{\tau_s(p)+1} \delta_{\lambda_s(p),0} \quad (9)$$

지금까지 설명한 관련된 식들은 참고문헌[1, 4]에서 상세히 설명되어 있다.

본 논문에서 고려하는 이와 같은 에이전트 모델 경제는 기본적으로 가격에 대해서 경쟁하고 판매자들은 위에서 설명된 소비자들에게 동일한 상품들을 제공하는 2인의 판매자들로 제한하며, 가격들은 불연속적이며, 최대 가격과 최소 가격 사이에서 결정된다. 또한 pricebot들은 차례대로 번갈아 가면서 그들의 가격을 결정하는 것을 가정한다.

2.2 Myoptimal Pricebot vs. Myoptmal Pricebot

myopically optimal, 또는 myoptimal 가격 결정 전략은 판매자의 기대 이익 π_s를 최대로 하는 최적 가격을 철저히 찾게 된다[5]. 그림 1에서는 두 개의 pricebot들이 모두 myoptimal 가격 결정 전략을 사용했을 때 전형적으로 발생되는 주기적인 가격 전쟁(cyclic price war)을 나타내고 있다.

그림 1에서 보는 바와 같이 myoptimal pricebot들은 최소 가격 지점 0.58에 도달 할 때까지 가격 이산 구간 ε만큼씩 서로 간의 제시 가격을 내리고 있다. 최소 가격 지점에 도달하게 되면 소비자 평가 값 $v = 1.0$으로 가격을 다시 결정함으로써 끊임없는 새로운 가격 전쟁 주기가 반복된다. 이러한 현상은 pricebot들이 기본적으로 예측 능력이 없는 myoptimal 가격 결정 전략을 기반으로 하고 있기 때문이다.

3장에서는 에이전트에게 장기적인 행동들에 대한 결과들을 예상할 수 있는 능력을 부여하는 방법을 소개하며 이러한 능력을 보유한 에이전트가 주기적 가격 전쟁을 줄이는 동시에 이익을 증가시킬 수 있음을 보인다.

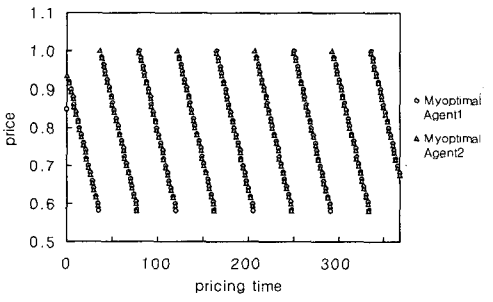

그림 1 Myoptimal pricebot들에 대한 가격 결정 변화

3. 다중 에이전트 Q-학습

3.1 Markov Decision Process와 강화 학습

2장에서 설명된 경제 모델 하에서 pricebot의 프로세스는 Markov decision process로 간주될 수 있다. Markov decision process는 다음과 같이 정의 된다.

정의. Markov Decision process는 튜플 <S, A, r, p>이며, 이때 S는 비 연속 상태 공간이며, A는 비 연

속 행동이고, $r : S \times A \rightarrow R$은 에이전트의 보상 함수이며, R은 실수의 집합이다. 또한 $p : S \times A \rightarrow \Delta$는 Δ가 상태 공간 S에 대한 확률 분포라고 할 때 변화 함수이다[6].

여기서 제안되는 시스템은 결정적(deterministic) Markov decision process이다. 에이전트는 현재의 상태에서 다음 행동을 선택하기 위해 정책 $\pi : S \rightarrow A$를 학습하게 된다. 임의의 정책 π에 의해서 획득 축적된 값(cumulative reward)은 다음과 같이 정의된다.

$$V(s,\pi) \equiv \sum_{t=0}^{\infty} \gamma^t E(r_t \mid \pi, s_0 = s) \quad (10)$$

단, s_0는 초기 상태이며, r_t는 시간 t에서의 보상이고, $\gamma \in [0,1]$는 할인율(discount factor)이다. 식 (10)은 다음과 같이 다시 기술 될 수 있다.

$$V(s,\pi) = r(s,a_\pi) + \gamma \sum_{s'} V(s',\pi) \quad (11)$$

단, a_π는 정책 π에 의해서 결정된 행동이고 s'는 다음 상태(succeeding state)이다. Markov decision process에서 에이전트의 목적은 모든 상태 s에 대해 $V(s, \pi)$를 최대로 하는 정책 π를 학습하는 것이다. 그러한 정책을 최적 정책(optimal policy)이라 하고 π^*로 나타낸다.

$$\pi^* \equiv \arg\max_{\pi} V(s,\pi), \ (\forall s) \quad (12)$$

어떤 $s \in S$에 대해 다음의 Bellman식에서와 같이 최적 정책 π^*가 존재함이 증명되었다.

$$V(s,\pi^*) = \max_a \left\{ r(s,a) + \gamma \sum_{s'} V(s',\pi^*) \right\} \quad (13)$$

이때 $V(s, \pi^*)$를 상태 s에 대한 최적 값(optimal value)이라고 한다. 실제 에이전트 경제에서는 에이전트들이 보상 함수(reward function)와 상태 변화 함수(state transition function)에 대한 완벽한 지식을 가지는 것은 어렵다. 그래서 에이전트는 이러한 함수들을 모르고서 환경과의 교류를 통해 직접적으로 최적 정책을 학습할 수 있는 방법이 필요하게 된다.

강화 학습(reinforcement learning)이란 그러한 문제를 풀 수 있는 강력하고 실제적인 방법이며, 환경 모델이 불필요한 이와 같은 비 모델 강화 학습 방법 중 하나가 Q-학습(Q-learning)이다. Q-학습의 기본 개념을 위해 식 (13)을 다음과 같이 표현할 수 있다.

$$Q(s,a) = r(s,a) + \gamma \sum_{s'} V(s',\pi^*) \quad (14)$$

즉 $Q(s, a)$는 상태 s부터 시작해서 첫 번째 행동으로 a를 적용하고, 그 후에 최적 정책을 따라가면서 획득될 수 있는 전체적으로 할인이 이루어진 축적된 보상(total discounted cumulative reward)을 의미하게 된다. 식 (13)과 (14)에 의해 다음과 같은 식을 기술할 수 있다.

$$V(s,\pi^*) = \max_a Q(s,a) \quad (15)$$

따라서 만일 $Q(s, a)$가 구해진다면 최적 정책을 발견할 수 있게 되는 것이다. 표 2에서는 Q-학습 알고리즘이 상세히 기술되어 있다. Q-학습의 수렴에 대한 증명에 관해서는 참고 문헌 [7]에 설명이 되어있다.

표 2 Q-학습 알고리즘

각각의 상태 s와 행동 a에 대해, 임의적으로 테이블의 요소 $\hat{Q}(s, a)$를 초기화한다;
현재 상태 s를 인식한다;
Do forever {
  행동 a를 선택하고 실행한다;
  즉시 보상(immediate reward) r을 취한다;
  새로운 상태 s'를 인식한다;
  $\hat{Q}(s, a)$에 대한 테이블 요소를 다음과 같이 갱신한다
  ($\hat{Q}$: 실제 Q값에 대한 평가 값);
  $\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$
  $s \leftarrow s'$;
}

3.2 다중 에이전트 Q-학습 알고리즘 설계

다중 에이전트 Q-학습 시뮬레이션을 위한 중요한 사항들에 대해 본 논문에서는 다음과 같이 정의한다[8].

- Q-테이블의 초기 값: immediate reward 값으로 설정
- 행동(action): 에이전트의 가격결정
- 상태(state): Q-학습된 에이전트에 의한 행동과 다른 에이전트에 의한 응답 행동
- Immediate reward: 위의 두 가지 행동에 대한 두 개의 보상들의 합

각각의 Q-값에 대해 학습하기 위해서, Q-학습을 하는 에이전트는 그 자신의 Q-값들을 위해 m개의 Q-테이블들을 유지해야 한다. 이때 m은 상태들의 총 개수다. 하나의 Q-테이블에는 $a^1 \in A^1$에 대응하는 행들과 $a^2 \in A^2$에 해당하는 열들로 구성된다. 여기서 A^1은 한 에이전트의 행동들의 집합이고 A^2는 다른 에이전트의 행동 집합이다. 그림 2에서는 위에서 설명한 Q-테이블을 그림으로 보여 주고 있다. 따라서 에이전트가 Q-학습을 위해 필요로 하는 전체 Q-테이블 요소의 수는 $m \times |A^1| \times |A^2|$이며, $|A^1|$와 $|A^2|$는 각각 A^1와 A^2의 크기이다.

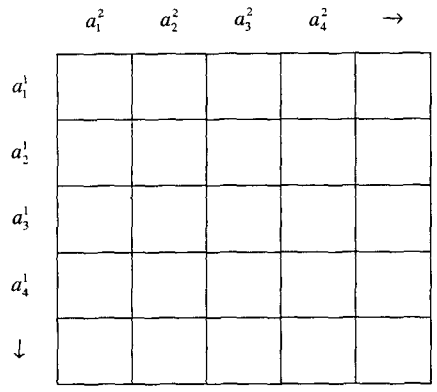

그림 2 Q-학습 에이전트의 하나의 Q-테이블 구조

3.3 Q-학습 Pricebot vs. Myoptimal Pricebot

Q-학습 에이전트가 근시안적 에이전트를 상대로 경쟁하게 될 때 Q-학습된 에이전트는 더욱 빨리 가격 전쟁(price war)을 포기하게 된다. 따라서 가격 전쟁의 폭이 줄어들게 되고 높은 평균 이익을 보장받게 된다[9]. Q-학습된 pricebot과 myoptimal pricebot에서의 가격 결정 결과가 그림 3과 표 3에 설명이 되어있다. Q-학습된 에이전트는 할인 매개변수(discount parameter) γ = 0.5로 트레이닝 되어있다. 그림 3에서 가격 전쟁의 모습이 에이전트들의 가격 (1.0, 1.0)에서 시작해서 Q-학습된 에이전트가 결정한 가격 0.78까지 지속된다. 이 가격에 대해서 myoptimal 가격 결정 전략을 가진 에이전트가 0.775로 제시를 하게 되면 Q-학습한 에이전트는 최소 가격 지점인 0.58로 가격을 급속히 하락시켜 결정하게 되고, 이는 myoptimal 에이전트로 하여금 소비자 상품 평가 값인 1.0으로 상승시켜 가격 결정을 하도록 하게 한다. 이와 같은 결과는 두 에이전트 모두 myoptimal 가격 결정 전략을 사용했을 때 나타나는 결과인 그림 1에 비해 가격 전쟁의 상황이 축소가 된다는 것을 의미한다.

표 3은 그림 3의 결과와 그림 1에서의 결과에서 얻어진 평균 이익을 나타낸다. 표 3에서 볼 수 있듯이 Q-학습된 에이전트와 myoptimal agent가 두 에이전트가 모두 myoptimal 가격 결정 전략을 사용할 때 보다 더 큰 평균 이익을 얻을 수 있음을 알 수 있다.

표 3 그림 1과 3의 simulation에서의 pricebot들의 평균 이익

Pricebots	평균 이익
(Myopic, Myopic)	(0.248, 0.248)
(Q learned, Myopic)	(0.329, 0.322)

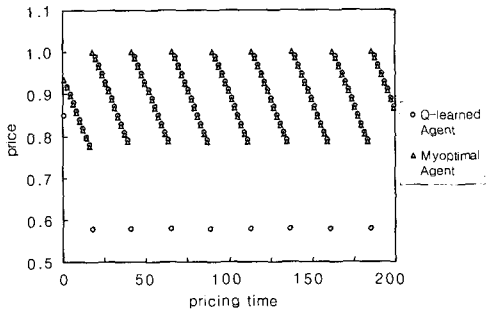

그림 3 Q-학습된 pricebot과 Myoptimal pricebot에 대한 가격 결정 변화

4. 범용적 온라인 Q-학습으로의 확장

4.1 Exploration과 Exploitation

다른 종류의 학습 방법에서는 찾아 볼 수 없는 강화 학습에서만 제기되는 문제가 바로 탐색(exploration)과 이용(exploitation)에서의 균형문제이다. '*Exploration*'은 모든 허용 가능한 상태-행동들이 Q-학습 수렴 법칙을 만족하기 위해 충분히 탐색되는 것을 보장하며, '*Exploitation*'은 greedy 정책을 적용함으로써 cost-to-go 함수를 최소로 하는 것을 추구하는 것이다[10].

강화 학습 에이전트는 보상을 얻기 위해 이미 알고 있는 행동을 이용해야 하지만, 이와 함께 앞으로 더 바람직한 행동의 선택을 위해서 탐색도 반드시 필요하게 된다. 그러나 임의의 단일 행동 선택에 있어서 exploration과 exploitation이 함께 이루어지는 것은 불가능하므로 exploration과 exploitation사이의 충돌(conflict)이 존재하게 된다. 따라서 강화 학습을 하는 에이전트에게 있어서 exploration과 exploitation의 균형을 이루는 것은 매우 중요하다. exploration이 이루어지는 것이 바람직한지 exploration이 이루어지는 것이 더 바람직한지는 평가 값, 불확실성, 잔존 활동의 수와 같은 복합적 요인에 근거하게 된다[11].

4.2 Mixed Nonstationary Policy

Q-학습 과정에서 최적 정책을 결정하기 위해 단일 에이전트는 일련의 상태-행동을 선택할 때 임의 상태-행동 선택 방법 (Uniform Random Selection, URS)을 사용할 수 있다[3, 8, 9]. 그러나 URS는 exploitation을 고려한 방법이라기 보다는 exploration에 더 중점을 둔 방식이기 때문에 최적 정책 학습을 위한 exploitation의 불확실성을 반영하게 된다. 강화 학습에서 중요하게 다루고 있는 exploration과 exploitation사이에서의 균형이란 관점에서 보았을 때 URS는 이러한 균형문제를 고려

하지 않은 방법이며, 그 결과 최적 값을 위해 각 상태에 해당하는 Q-테이블을 접근하는 회수가 크게 증가하게 된다. 따라서 URS는 비록 Q-학습이 온라인 트레이닝을 위해 사용될 수 있는 학습 방법이라고는 하나 실 세계 환경에서 범용적 온라인 학습에는 부적합하게 된다.

이러한 강화 학습에서의 exploration과 exploitation사이의 균형 문제를 해결하고, 동시에 Q-table을 접근하는 회수를 줄임으로써 실 세계에서 범용적 온라인 Q-학습이 가능하도록 하기 위해 본 논문에서는 Mixed Nonstationary Policy (MNP)를 제안한다. MNP는 [12]에 근거를 두며 위에서 보여주었던 shopbot과 pricebot 에이전트 경제 모델에서의 Q-학습 에이전트를 위한 시뮬레이션에 적합하도록 크게 다음과 같이 수정되었다.

- 본 논문에서의 Q-학습은 결정적(deterministic) 환경을 적용하므로 보조 마르코프 프로세스(auxiliary Markov process)와 원형 마르코프 프로세스(original Markov process)의 상태 변화 확률은 고려하지 않는다.
- Q-학습 동안 auxiliary Markov process는 일련의 상태-행동 선택 시 균일 임의 선택 방식에 기초한다.

MNP는 auxiliary Markov process와 Q-학습에 의해 결정되는 stationary greedy 정책을 통해서 통제가 이루어지는 original Markov process의 전환을 수행한다. 이와 같은 복합정책은 먼저 auxiliary process의 어느 한 상태에서 행동들을 선택하며, 그리고서 original controlled process로 이동하는 일련의 전환을 그림 4처럼 반복하게 된다. auxiliary process는 단계 L이라는 일정한 수 동안 프로세스를 수행하게 되며 original controlled process의 수행 시간은 매 전환마다 $kL(k = 1,2,3,\cdots)$만큼씩 증가하게 된다.

MNP가 적용된 Q-학습 에이전트는 original controlled process의 실행 시간 동안 Q-학습에 의해 결정되는 stationary greedy 정책, 즉 현 단계의 해당 상태에서 여러 가능한 행동들 중 greedy 행동을 선택하는 방식으로 학습함으로써 auxiliary Markov process와 original controlled process에 의해 평가 측정된 최적 정책에 대해 1의 확률로 exploitation이 이루어질 수 있도록 하며, URS에서 발생하는 최적 정책을 위한 exploitation의 불확실성의 문제를 해결하게 된다. 그 결과 MNP가 적용된 에이전트는 URS를 사용한 에이전트에 비해 보다 적은 Q-테이블의 접근 회수로 더 빠르게 최적 값에 수렴함으로써 범용적인 온라인 Q-학습이 가능하게 된다.

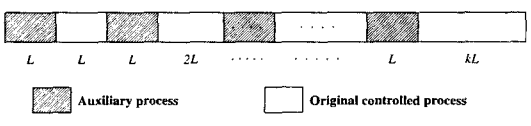

그림 4 시간에 따른 MNP의 프로세스 전환

5. 실험 및 성능평가

최적 값으로의 수렴을 위해 MNP를 사용하는 Q-학습과 URS를 사용하는 Q-학습의 성능 비교를 위해, 본 실험은 자바로 프로그래밍 되었으며 MNP의 구현을 위해 멀티쓰레딩 기법이 사용되었다. 본 논문에서의 모든 실험들은 동일한 입력에 대해 각각 5회 반복 실험을 하였고, 그 결과들을 각각의 입력에 대해 평균하였다. 표 4에는 실험을 위해 사용된 환경 변수 값들이 나타나 있다.

그림 5에는 에이전트들이 URS와 MNP를 적용했을 때 Q-테이블 $Q(1.0, 0.995)$에서 $Q(0.775, 0.770)$까지, 최적 Q-값으로 처음으로 도달했을 때의 각각의 Q-테이블 접근 회수를 보여주고 있다. 그림 5에서 볼 수 있듯이 거의 모든 Q-테이블에서 MNP가 적용된 Q-학습 에이전트가 URS를 사용한 Q-학습 에이전트보다 적은 접근 회수를 가지고 빠르게 최적 Q-값으로의 수렴이 시작됨을 알 수 있다. 이때 Q-테이블 $Q(0.780, 0.775)$에서만 URS의 성능이 MNP의 성능보다 좋음을 발견할 수 있다. 그러나 MNP에서 $Q(0.780, 0.775)$의 평균 접근 회수는 224.4회, URS에서 동일한 테이블의 평균 접근 회수는 204.6회로 그 차이는 20회 이하의 적은 회수이므로 $Q(0.780, 0.775)$에서만 발생된 이러한 예외 현상은 전체적인 학습 성능에는 크게 영향을 끼치지 않는다.

그림 6에서는 그림 5에서 나타내고 있는 MNP를 적용한 Q-학습과 URS가 사용된 Q-학습에서의 최적 Q-값으로의 수렴이 시작되는 전체적인 평균 접근 회수를 보여준다. 그림 6에서 URS가 사용되었을 때는 평균 접근 회수가 약 1065.1회, MNP를 적용했을 때는 약

표 4 실험에서 사용된 환경 변수

환경 변수	값	의 미
v	1.0	소비자의 상품 평가 값
c	0.5	판매자의 생산 비용
w_s	0.25	*Any Seller* 전략을 사용하는 소비자 분포
w_b	0.75	*Bargain Hunter* 전략을 사용하는 소비자 분포
γ	0.5	할인 매개변수
ε	0.005	가격 이산 구간
p_{min}	0.58	판매자의 최소 제시 가격
L	100.0	MNP에서의 일정 수의 단계

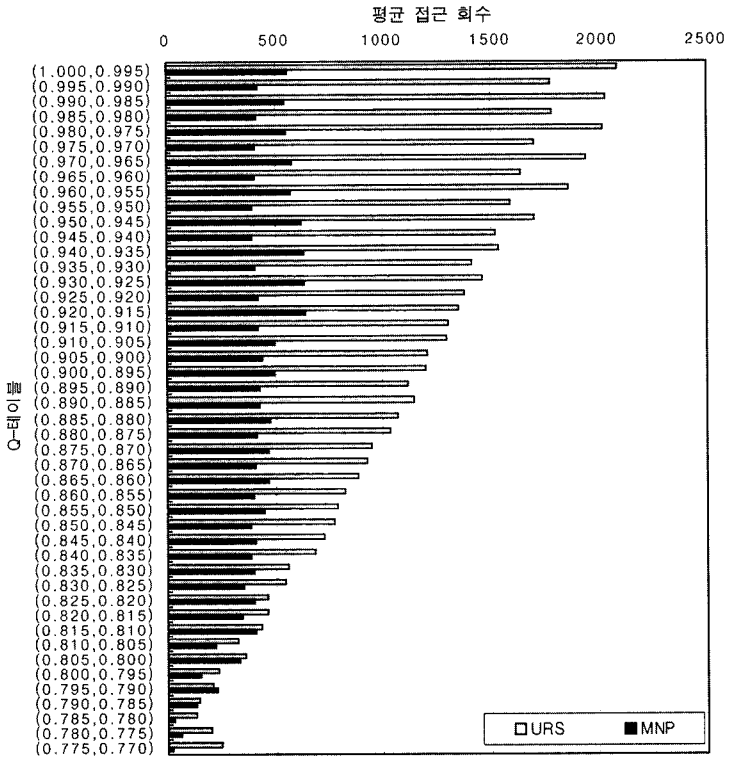

그림 5 Q-테이블에서 최적 Q-값으로의 수렴이 시작 되는 평균 접근 회수

404.8회로 MNP가 적용된 Q-학습 pricebot이 URS가 적용된 Q-학습 pricebot에 비해 더 적은 Q-테이블의 접근 회수로 약 2.6배 더 빠른 최적 정책의 선택이 가능함을 보이고 있다.

위의 실험 결과들로부터 가격 결정 알고리즘으로 Q-학습을 사용하는 에이전트에 MNP를 적용시킬 경우 URS를 사용하는 경우에 비해 Q-학습에서 적은 Q-테이블의 접근 회수로 더 빠른 최적 정책으로의 수렴을 가능하게 함으로써 온라인 상에서의 가격 결정을 위한 범용적 Q-학습이 가능함을 알 수 있다.

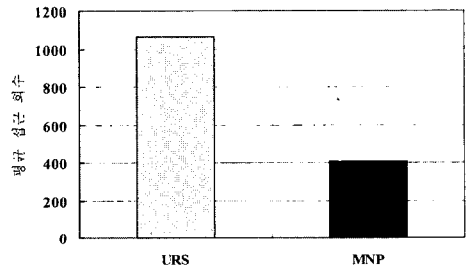

그림 6 최적 Q-값으로의 수렴 발생을 위한 전체 평균 접근 회수

6. 결론 및 향후 연구

에이전트 기반의 경제 모델에서 판매자들의 수익성을 보장해 주기 위한 pricebot이라고 불리는 에이전트가 가격 결정 알고리즘으로 Q-학습을 사용하게 될 때 Q-학습의 수렴을 위한 일련의 상태-행동을 선택해야 한다. 이를 위해 에이전트가 최적의 정책에 대한 exploitation의 불확실성을 반영하게 되는 URS를 사용하게 될 때 각각의 특정 상태에 대한 Q-테이블을 접근하는 회수가 증가하게 됨으로써 URS는 온라인상에서의 일반적 학습에 부적절하다. 이러한 문제를 해결하기 위해 MNP가 적용된 Q-학습이 본 논문에서 제안되었으며 이를 통해 URS가 Q-학습 에이전트에 적용되었을 때 발생되었던 강화 학습에서의 exploration과 exploitation의 균형 문제를 해결함으로써 각 상태에 해당하는 Q-테이블로의 적은 접근 회수로 빠른 Q-학습의 수렴을 가능하게 하여 실 세계 온라인 환경 하에서 적절하게 학습될 수 있음을 보였다.

본 논문과 관련된 향후 연구에 있어서 하나의 에이전트가 학습을 위해 필요로 하는 공간은 참여하는 에이전트의 수에 지수적으로 증가하므로 다수의 에이전트들로 구성되는 경제 모델을 위해 보다 작은 행동 공간을 사용하는 방안에 대한 연구가 필요하다. 또한 실 세계에서 가능한 상태와 행동의 수가 아주 클 수 있으며 이러한 환경에서는 lookup 테이블에 의한 표현 방식은 적절한 방법이라고 할 수 없다. 따라서 이를 위해 온라인 환경을 고려한 효율적인 함수 근사자(function approximator)와 강화 학습 방법과의 연동에 대한 연구가 이루어져야 할 것이다.

참 고 문 헌

[1] A. Greenwald and J. O. Kephart, "Shopbots and Pricebots," *Proc. Int'l J Conf. Artificial Intelligence*, Stockholm, Sweden, 1999.

[2] C. J. C. H. Watkins, "Learning from delayed rewards," *Ph. D. thesis*, Cambridge University, 1989.

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

[4] A. Greenwald, J. Kephart, and G. Tesauro, "Strategic Pricebot Dynamics," *Proc. 1st ACM Conf. Electronic Commerce*, Oct. 1999.

[5] G. J. Tesauro and J. O. Kephart, "Foresight-based pricing algorithms in an economy of software agents," *Proc. ICE-93*, 1998, pp. 37-44.

[6] J. Hu and M. P. Wellman, "Multiagent reinforcement learning: theoretical framework and an algorithm," *Proc. Int'l Conf. Machine Learning*, 1998.

[7] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997, pp. 378-379, p. 382

[8] M. Sridharan and G. Tesauro, "Multi-agent Q-learning and regression trees for automated pricing decisions," *Proc. 17th Int'l Conf. Machine Learning, Stanford*, CA., 2000.

[9] G. Tesauro, "Pricing in agent economies using neural networks and multi-agent Q-learning," *Proc. IJCAI-99 Workshop, Learning About, From and With Other Agents*, Stockholm, Sweden, Aug. 1999.

[10] S. Haykin, *Neural Network*, 2ndEd, Prentice-Hall, New Jersey, 1999, p. 625.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press/Bradford Books, 1998, pp. 4-5, 26-27.

[12] G. Cybenko, R. Gray and K. Moizumi, "Q-Learning: A Tutorial and Extensions," *Proc. Conf. Mathematics of Artificial Neural Networks*, Oxford University, England, July, 1995.

박 찬 건

2001년 경성대학교 컴퓨터공학 학사
2003년 연세대학교 컴퓨터과학 석사
2003년~현재 팬택&큐리텔 중앙연구소 CDMA단말기 소프트웨어 연구원. 관심분야는 인공지능, 지능형 에이전트, 전자상거래

양 성 봉

1981년 연세대학교 공학사. 1984년 Univ. of Oklahoma 컴퓨터과학 석사 1992년 Univ. of Oklahoma 컴퓨터과학 박사. 1993년~1994년 전주대학교 전자계산학과 전임강사. 1994년~현재 연세대학교 컴퓨터산업공학부 부교수. 관심분야는 전자상거래, 그래픽스, 인터넷 컴퓨팅