12

강화 학습과 감독 지식의 융합기술

서강대학교 ▌김성완 · 장형수*

1. 서 론

당신이 지금 한 프로야구 경기에 투수로 등판해 있다고 가정해 보자. 당신 앞에 서 있는 타자에게 어떤 구질의 공을 던지느냐에 따라서 출루 상황과 점수, 상대 팀의 전술이 바뀌게 될 것이다. 그리고 이러한 상황의 변화는 당신에게 유리하게, 혹은 불리하게 작용할 것이고 다음 피칭에서 던지게 될 구질의 종류에도 영향을 미칠 것이다. 이처럼 당신은 매 피칭마다 어떤 공을 던질지 선택을 해야 하고, 순차적인 선택들은 결국 당신의 팀의 승패를 좌우할 것이다. **지금 당신이 처한 문제 상황, 즉 시간 스텝에 따라서 행동을 결정하고 그로 인한 상황의 변화로부터 피드백을 받는 문제 상황을 순차적 의사결정문제(sequential decision problem)라 한다.**

순차적 의사결정 문제는 로봇과 같이 지능을 가진 에이전트가 자주 접하게 되는 문제이며, 널리 알려져 있듯이 이는 마르코프 의사결정과정(Markov decision processes, MDPs)으로 형식화될 수 있다[1]. 만일 MDP 모델의 파라미터들을 에이전트가 모두 알고 있다면, 잘 알려진 해법인 value iteration(VI), policy iteration(PI), 또는 linear programming[2]을 사용하여 주어진 optimality criterion에 대한 "optimal" (혹은 approximately optimal) "policy"를 구할 수 있다. 하지만 일부 파라미터들을 에이전트가 모를 때, 즉 hidden MDP process를 당면하고 있는 **에이전트는 매 결정시간 스텝마다 직접 행동을 선택함으로써 얻어지는 상황 정보(state information)와 강화 신호(reinforcement signal, 보상/reward)들만을 이용하여 optimal policy를 학습해야 한다. 강화 학습(reinforcement learning, RL)[1] 알고리즘은 optimal policy를 학습하는데 널리 사용된다.** 강화 학습과 다른 패러다임의 학습 알고리즘으로 감독 학습(supervised learning)[3]이 있다. "선생(teacher)"과 함께 학습하는 알고리즘[10]이 감독 학습의 한 예로서, 일반적으로 optimal policy에 느리게 수렴한다는 단점이 있다.

많은 문제 상황에서 학습자의 학습 효율을 높이기 위해 선생 혹은 감독자가 학습자에게 직접 또는 간접적인 정보를 제공한다. 이 현상은 사람들의 일상적인 삶에서 자주 보이는 모습이다. 이에 착안하여 그동안 학습과 연관된/학습된 특정한 사전지식 혹은 "감독된" 지식을 강화 학습의 과정에 혼합시켜 에이전트의 학습속도를 향상시키려는 몇몇 연구가 진행되어 왔다[11~14]. 하지만 어떤 연구에서도 학습이론에 있어 가장 중요한 학습 과정에서의 수렴 속성(convergence property)에 대한 언급은 없었다.

본고에서는 강화 학습이 다수 학습(multiple learnings), 그리고 전문가의 조언(expert advice)과 조합되어 실제로 학습의 수렴 속도를 향상시킨 새로운 학습 프레임워크[8]를 소개하고자 한다. 본고에서 언급하는 전문가(expert)는 학습자, 즉 에이전트의 의사결정에 도움을 주는 모든 시스템을 일컫는다. 다수 학습은 SARSA(0)[1]를 사용하는 하나의 기본 에이전트(base-Agent)와, 각각 다른 학습 방법을 사용하는 여러 개의 서브에이전트(subagents)들이 동시에 학습함으로써 이루어진다. 이 과정에서 Ng et al[7]의 논문에서 소개된 "potential-based reinforcement function"을 사용하여 서브에이전트의 학습결과가 기본 에이전트의 학습에 반영될 수 있도록 기본 에이전트의 강화 신호를 조성(shaping)하였다. 서브에이전트들은 각각 다른 학습방법을 동시에 수행함으로써, 혹은 가능하다면 전문가의 조언도 적용함으로써 얻어진 값들을 기본 에이전트의 입력 값에 사용한다. 기본 에이전트는 서브에이전트로부터의 입력 값들을 "potential-based reinforcement function" 값으로 합병(merge)하고 이를 SARSA(0)의 Q-함수 업데이트 공식에 적용한다. 이처럼 "감독"된 SARSA(0)는 optimal policy에 상대적으로 보다 빠르게 수렴한다.

Russell과 Zimbers 역시 [4]에서 다중 학습에 대해 위와 관련된 연구 결과를 선보인 바 있다. 그들은 특

* 정회원

별한 보상(reward) 구조를 생각해 내었는데, 주어진 보상 함수(reward function)를 여러 개의 부보상 함수(subreward function)로 분할하여 서브에이전트들이 각각의 subreward 함수에 의하여 학습하고 그 학습 결과들을 마스터에이전트(master Agent)가 조합하여 사용한다는 것이다. 하지만 본고에서 소개할 "감독 학습/지식과 강화 학습이 조합된 학습 방법"은 서브에이전트들이 학습하는데 있어 보상 함수에 대한 어떠한 가정도 없다는 점에서 [4]의 연구와 근본적인 차이점이 있다.

학습을 통하여 구해진 policy를 저장하여 앞으로의 학습에 재사용하는 policy reuse기법 역시 학습 속도를 향상시키는 기법 중의 하나이며 최근 활발하게 연구되고 있는 분야이다. 본고에서는 policy reuse기법을 효과적으로 사용한 알고리즘을 제시한 Fernàndez와 Veloso[9]의 논문을 언급할 것이다. 그들은 Exploration-Exploitation 전략 [1]의 하나로 널리 사용되고 있는 ε-greedy 기법에 policy reuse를 결합하여 일정 확률로 ε-greedy 기법을, 나머지 확률로 policy reuse기법을 사용하는 p-reuse exploration 전략을 소개하였다. 또한 에이전트가 놓인 상황과 현재 저장된 policy가 학습되었을 때의 상황을 비교하여 그 유사성(similarity)에 따라 π-reuse exploration을 사용하는 "PRQ-학습 알고리즘"과, 이를 통해 policy들을 저장하는 policy 라이브러리를 생성 및 유지하는 "PLPR 알고리즘"을 제시하였다. 로봇 내비게이션 실험을 통하여 그들은 자신들이 논문에서 제시한 알고리즘이 기존의 학습 알고리즘의 성능을 한 단계 향상시킴을 보여주었다.

본고는 다음과 같은 형식으로 구성되어 있다. 2절에서는 강화 학습에 대한 기본적인 배경 지식을 제공할 것이다. 이는 뒤에 소개할 학습 프레임웍을 이해하는데 많은 도움이 될 것이다. 3절에서는 다중 학습과 전문가의 조언을 결합한 새로운 융합된 학습 프레임웍을 보다 자세히 소개할 것이다. 4절에서는 policy reuse기법에 대해서 보다 자세히 소개하고, 3절에서 소개한 프레임웍과 policy reuse기법과의 연관성과 그 결합 가능성에 대해 논의할 것이다. 5절에서 본고를 통하여 이야기하고자 한 내용의 결론을 제시할 것이다.

2. Backgrounds

학습 에이전트는 MDP 모델로 표현되는 환경과 상호작용한다. 4개의 튜플을 가진 어떤 MDP $M=(X, A, P, R)$이 있다고 하자. X는 환경 안에 존재하는 상태(state)들의 유한집합이고, A는 행동(action)들의 유한집합이다(모든 행동은 모든 상태 내에서 가능하다고 가정한다). P는 집합 $\{(x, a) \mid x\in X,\ a\in A\}$를 X에 가능한 모든 확률분포로 mapping하는 상태전이함수이다. 상태 x에서 어떤 행동 a를 선택하여 상태 y로 전이할 수 있는 확률을 $P(y|x, a)$라 하자. R은 $X\times A\times X$를 실수집합 R로 mapping하는 보상함수라 하고, 상태 x에서 어떤 행동 a를 선택하여 상태 y로 갔을 때의 보상(reward)을 $R(x, a, y)$라 하자.

***Policy* π는 X에서 A로의 mapping 함수로 정의되며 Π를 모든 가능한 policy들의 집합이라고 할 때, 초기 상태 x에서 $\pi\in\Pi$인 policy π에 따라 행동을 택했을 때 얻어지는 값 $V^{\pi}(x)$를 다음과 같이 정의한다:**

$$V^{\pi}(x)=E\left[\sum_{t=0}^{\infty}\gamma^{t}R(X_t,\pi(X_t),X_{t+1})\,|\,X_0=x\right] \qquad (1)$$

X_t는 시간 t에서의 상태를 나타내는 random variable이며, $\gamma\in(0, 1)$은 고정된 discount factor이다.

$V^{*}(x)=\max_{\pi\in\Pi}V^{\pi}(x)$, $x\in X$라 하자. 그리고 $V^{*}(x)$를 상태 $x\in X$의 *optimal value*라고 정의하자. 다음의 결과들은 Bellman's optimality principle에 의한 것이며 이는 [4]에 잘 나타나 있다.

Theorem 1 : 모든 $x\in X$에 대해서,

$$V^{*}(x)=\max_{a\in A}\left\{\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma V^{*}(y)\big)\right\} \qquad (2)$$

이고 $V^{*}(x)$, $x\in X$는 오직 하나의 값을 가지며 optimal policy π^{*}는 $V^{\pi^{*}}(x)=V^{*}(x)$를 만족하는 모든 $x\in X$에 대해 다음과 같이 정의된다.

$$\pi^{*}(x)\in\arg\max_{a\in A}\left\{\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma V^{*}(y)\big)\right\} \qquad (3)$$

$X\times A$에 대한 Q^{*}-함수를 $x\in X$와 $a\in A$에 대해서 다음과 같이 정의하자:

$$Q^{*}(x,a)=\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma V^{*}(y)\big) \qquad (4)$$

그러면, Q^{*}는 $x\in X$와 $a\in A$에 대해서 다음의 fixed-point equation을 만족한다.

$$Q^{*}(x,a)=\sum_{y\in X}P(y|x,a)\big(R(x,a,y)+\gamma\max_{a'\in A}Q^{*}(y,a')\big) \qquad (5)$$

에이전트의 학습이란, 바로 MDP 모델 M에서 상태전이함수 P와 보상 함수 R을 알지 못할 때 Q^{*}-함수를 학습하는 것을 말한다.

확률적 근사(stochastic approximation)은 fixed-point equation을 푸는 잘 알려진 방법이다[5]. Watkins에 의해 발전된 Q-learning[1] 역시 확률적 근사를 기반으로 한 접근으로 (5)를 푸는 기법이다[5]. Q-learning은 exploration-exploitation 전략에 영향을 받지 않는다는 점에서 *off-policy* 학습이다[1]. 에이전트가 사용하는 exploration-exploitation 전략이 각각의 모든 상태를 무한하게 방문하고, 각 상태의 모든 행동이 무한하게 선택되었을 때(explored) Q-learning은 $V^*(x) = \max_{a\in A} Q^*(x, a)$, $x\in X$인 벡터 Q^*에 수렴한다. **Q-learning과 같은 *off-policy* 학습이 학습 전략(exploration-exploitation 전략)에 대해 어떤 속성도 요구하지 않는 반면에, *on-policy* 학습은 학습 전략에 의해 선택된 행동을 기반으로 하여 Q-value를 갱신한다. 즉 *on-policy* 학습에서 optimal policy로의 수렴은 exploration-exploitation 전략에 의해 영향을 받게 된다.**

SARSA(0)[1][6]는 *on-policy* 학습 방법이며 "다음" 시간 단위에서 관찰되는 값만을 사용하여 현재 상태의 Q-값을 갱신한다. SARSA(0)는 3절에서 소개될 다중 학습에서 기본 에이전트가 사용하게 될 학습 방법이기 때문에 이 절에서 자세히 소개될 것이다.

비동기(asynchronous) SARSA(0)에서는 각각 discrete한 시간 스텝 $t\geq 0$에서 학습 에이전트가 상태 x_t를 관찰(observe)하고 학습 전략(learning strategy) ϕ_t에 따른 x_t에서의 행동 a_t를 선택한다. 그리고 확률적으로 결정되는 다음 상태 $x_{t+1}(P(y|x_t, a_t))$인 y와 같다)를 관찰하고 ϕ_t에 따라 x_{t+1}에서의 행동 a_{t+1}을 선택한다(하지만 실제로 행하지는 않는다). 이러한 과정을 통해 얻어지는 값들을 이용하여 $X\times A$ 집합으로 정의된 Q_t 벡터의 $Q_t(x_t, a_t)$-부분을 다음의 식에 의해 갱신한다.

$$\begin{aligned} Q_{t+1}(x_t, a_t) \leftarrow & Q_t(x_t, a_t) \\ & + \alpha_t(x_t, a_t)[R(x_t, a_t, x_{t+1}) \\ & + \gamma Q_t(x_{t+1}, a_{t+1}) - Q_t(x_t, a_t)] \end{aligned} \tag{6}$$

$\alpha_t(x_t, a_t)$는 음의 값을 갖지 않는 스텝크기(stepsize)를 나타내는 계수이며, 모든 $(x, a) \neq (x_t, a_t)$ 쌍에 대해 0의 값을 갖는다.

이처럼 시간 t에서 오직 $Q_t(x_t, a_t)$만을 비동기적으로(asynchronously) 갱신한다. 앞으로 소개될 절들에서 우리는 ε_t-greedy 전략 [6]을 시간 t에서의 ϕ_t로 사용할 것이다. 다시 말해 시간 t에서 $1-c/n_t(x_t)$의 확률로 ($c\in(0, 1)$) greedy한 행동 $a_t\in \arg\max_{a\in A} Q_t(x_t, a)$가 선택되며, $c/(|A|\, n_t(x_t))$의 확률로 $a_t = a$, $a\in A$가 된다. $n_t(x_t)$는 시간 스텝 t까지의 상태 x_t로의 방문 회수를 의미한다.

Singh et al.[6]은 M이 communicating하다는 가정 아래 ϕ_t가 optimal policy π^*로, Q_t가 Q^*로 $t\to\infty$일 때 수렴함을 보였다(위의 문제 설정을 따랐을 때). 어떤 MDP가 communicating하다는 것은 policy 집합 Π에서 어떠한 policy를 선택하여 고정함으로써 얻어지는 어떠한 Markov Chain에서라도 어떠한 상태라도 다른 어떤 상태로부터 도달할 수 있음을 의미한다.

3. Potential-based 융합 기술

3.1 강화 학습들의 조합

MDP $M=(X, A, P, R)$이 주어져 있다고 할 때 에이전트는 Q^*_M-함수를 학습하고자 할 것이다(이제부터 함수의 아래첨자 M은 그 함수가 M에 관련된 것임을 의미하는 것으로 사용한다). 다음의 MDP $M'=(X, A, P, R')$을 생각해 보자. **M'는 M으로부터 *potential* function $\Phi : X\to R$에 의해서 $x, y\in X$, $a\in A$에 대해 다음과 같이 변환되었다.**

$$R'(x, a, y) = R(x, a, y) + \gamma\Phi(y) - \Phi(x) \tag{7}$$

위의 식에 사용된 $X\times X$에 대한 함수 $F(x, y) = \gamma\Phi(y) - \Phi(x)$, $x, y\in X$를 *potential based reinforcement function*이라고 한다. Ng et al.[7]은 모든 $x\in X$, $a\in A$에 대해 다음이 성립함을 보였다.

$$Q^*_{M'}(x, a) = Q^*_M(x, a) - \Phi(x), \text{ and}$$

$$V^*_{M'}(x, a) = V^*_M(x, a) - \Phi(x)$$

이는 어떤 potential 함수 $\Phi : X\to R$에 대해서도 성립한다. 따라서 M'에 대한 optimal policy는 M에 대한 optimal policy이기도 하며, 결국 $Q^*_{M'}$-함수를 학습하는 것과 Q^*_M-함수를 학습하는 것은 equivalent하다.

다중 학습에서 기본 에이전트(base-agent)는 위의 potential function을 사용하여 서브에이전트(subagent)들의 학습을 반영한다. 모든 서브에이전트들과 기본 에이전트는 비동기적인 자신들만의 시간스텝을 갖는다. 기본에이전트의 현재 갱신(update) 시간이 t일 때 서브에이전트 i의 갱신 시간을 t_i라고 하자. 그리고 Q_{t_i}를 서브에이전트 i의 t_i에서의 Q^*_M-함수에 대한 estimate값이라고 하자. 기본에이전트는 SARSA(0) (6)에서의 갱신 룰을 따르되 시간 t에서의 ϕ_t에 대해 ε_t-greedy 전략을 사용하며, MDP M'에 대한 학습을 하게 된다. 그 갱신 룰은 다음과 같다.

$$\begin{aligned} Q_{t+1}(x_t, a_t) \leftarrow & Q_t(x_t, a_t) \\ & + \alpha_t(x_t, a_t)[R(x_t, a_t, x_{t+1}) \\ & + \gamma\Phi(x_{t+1}; t_1, \ldots, t_m) - \Phi(x_t; t_1, \ldots, t_m) \\ & + \gamma Q_t(x_{t+1}, a_{t+1}) - Q_t(x_t, a_t)] \end{aligned} \tag{9}$$

이는 (6)의 식에 nonstationary(시간에 의존적인) potential-based reinforcement function $\gamma\Phi(x_{t+1}; t_1, \cdots, t_m) - \Phi(x_t; t_1, \cdots, t_m)$이 추가된 것이다. 여기서 Q_t는 $Q_t(x_t, a_t; t_1, \cdots, t_m)$와 같이 표기되어야 하지만 간단한 표기를 위해 $(t_1, \cdots, t_m)$을 생략하였다. 함수 Φ는 다음과 같이 정의된다.

$$\Phi(x_t; t_1, \ldots, t_m) = \sum_{a\in A}\left(\frac{1}{m}\sum_{i=1}^{m} Q_{t_i}^i(x_t, a_t) \times \theta(x_t, a; t_1, \ldots, t_m)\right) \quad (10)$$

여기서 $\theta(x_t, a; t_1, \cdots, t_m)$는 다음과 같이 주어진다.

$$\theta(x_t, a; t_1, \ldots, t_m) = \frac{\sum_{i=1}^{m} I\left(a\in \arg\max_{b\in A} Q_{t_i}^i(x_t, b)\right)}{\sum_{a'\in A}\sum_{i=1}^{m} I\left(a'\in \arg\max_{b\in A} Q_{t_i}^i(x_t, b)\right)} \quad (11)$$

또한 $\alpha_t(x_t, a_t) = 1/n_t(x_t, a_t)$, $x_t\in X$, $a_t\in A$로, 그리고 $(x, a) \neq (x_t, a_t)$인 모든 (x, a)을 0으로 한다. 함수 I는 $a\in A$일 때 $a\in \arg\max_{b\in A} Q_{t_i}^i(x_t, b)$이 참일 경우 $I\{a\in \arg\max_{b\in A} Q_{t_i}^i(x_t, b)\} = 1$, 그 이외에는 0을 갖는 것으로 주어져 있다.

Potential function Φ를 위와 같이 정한 이론적 근거는 다음과 같다. θ-함수는 행동 a가 optimal한 행동일 확률에 대한 추정치이다. 각 서브에이전트들은 어떤 행동이 optimal한 행동인지를 각자 다른 서브에이전트들과 독립되어 수행되는 학습 단계마다 "표현"해 주며, 이러한 서브에이전트들의 축적된 정보로부터 위의 추정치가 구하여지는 것이다. 또, 우리는 $Q^*_M(x_t, a)$에 대한 추정치로 $Q_{t_i}^i(x_t, a)$, $i = 1, \cdots, m$의 평균을 사용하며, $V^*_M(x_t)$에 대한 추정치로 현재 $Q^*_M(x_t, a)$의 추정치의 θ에 대한 가중 평균값을 사용한다. 즉 (10)에서 함수 $\Phi(x_t)$는 $V^*_M(x_t)$의 추정치를 의미한다. 확률 추정치 θ는, 모든 i에 대해서 $x\in X$, $a\in A$일 때 $\lim_{t_i\to\infty} Q_{t_i}^i(x, a) = Q^*_M(x_t, a)$라는 가정 하에 0이 아닌 값으로 수렴하며, 이와 같은 사실들은 다음의 수렴 결과를 성립시킨다.

Theorem 2 : $M = (X, A, P, R)$이 communicating하고, potential $\Phi(x) = V^*_M(x)$, $x\in X$일 때 M으로부터 변환된 MDP $M' = (X, A, P, R')$이 있으며, R'는 (7)에 의해 정의되었다고 하자. 그러면, 업데이트 공식 (9)의 ϕ_t로 ε_t-greedy를 사용하는, M'에 대한 기본에이전트의 SARSA(0)에 대해 다음이 성립한다.

$$\lim_{t\to\infty}\lim_{\forall i, t_i\to\infty} Q_t(x, a; t_1, \ldots, t_m) = Q^*_{M'}(x, a), \quad x\in X, \ a\in A$$

그리고 ϕ_t는 M에 대한 optimal policy π^*_M로 수렴한다.

3.2 전문가의 조언들(expert advices)의 조합

$k = 1, \cdots, I$명의 전문가들이 있으며 각각은 기본에이전트에게 각 상태에서 행동들의 집합 A에 대한 확률 분포의 형태로 명시적인 조언을 해준다고 가정하자. 여기서 전문가는 CBR(case based reasoning), neural-network learning, IBL(instance based learning), decision tree learning[9], stochastic learning automata[10], model-based / model-free RL[11][12] 등 학습 결과로 모든 행동들에 대한 확률 분포를 가져오는 메커니즘이 될 수 있다. 강화 학습 방법을 사용하는 서브에이전트들과는 달리 전문가는 행동을 확률 분포의 형태로 추천하기 때문에 V^*_M값에 대한 어떠한 추정치를 제공하지 않는다.

만일 에이전트가 상태 x에 대해 어떠한 조언도 가지고 있지 않다면, 행동 집합 A에 대해 동일한 확률 분포를 취할 것이다. 전문가 k가 그의 로컬 시간 s_k에서 제공해 주는, 상태 $x\in X$에서 행동 $a\in A$가 optimal한 행동일 확률을 $\rho^k_{s_k}(x, a)$라 하자.

$m \geq 1$인 경우, 즉 적어도 하나 이상의 강화 학습을 사용하는 서브에이전트가 있다고 하자. 이제 potential function $\Phi(x; t_1, \cdots, t_m)$을 $\Phi(x; t_1, \cdots, t_m, s_1, \cdots, s_l)$으로 확장하고, 상태 x_t에서 행동 a의 optimality에 대한 확률의 추정치 $\theta(x_t, a; t_1, \cdots, t_m)$을 $\theta(x_t, a; t_1, \cdots, t_m, s_1, \cdots, s_l)$로 확장하고 다음과 같이 정의한다.

$$\theta(x_t, a; t_1, \ldots, t_m, s_1, \ldots, s_l) \leftarrow \theta(x_t, a; t_1, \ldots, t_m) \times \frac{\sum_{k=l}^{l} \rho^k_{s_k}(x_t, a)}{\sum_{a'\in A}\sum_{k=l}^{l} \rho^k_{s_k}(x_t, a')}$$

$l = m$일 경우 다음과 같은 확장도 가능하다.

$$\theta(x_t, a; t_1, \ldots, t_m, s_1, \ldots, s_l) \leftarrow \frac{\sum_{k=1}^{m} I\left(a\in \arg\max_{b\in A} Q_{t_i}^i(x_t, b)\right)\rho^k_{s_k}(x_t, a)}{\sum_{a'\in A}\sum_{k=1}^{m} I\left(a'\in \arg\max_{b\in A} Q_{t_i}^i(x_t, b)\right)\rho^k_{s_k}(x_t, a')}$$

θ의 확장에 위의 방법만 존재하는 것은 아니다. 감독된(supervised) 입력 $\rho^k_{s_k}$가 모든 k에 대해 $s_k\to\infty$일 때, 시간에 대해 독립적인(stationary) 확률 분포로 수렴하면 Φ 역시 stationary한 함수로 수렴하게 된다. 이는 기본 에이전트의 SARSA(0)의 수렴을 보장한다. $m = 0$일 때, 즉 서브에이전트가 없을 때에는, 전문가의 조언을 적용하여 potential-based reinforcement func-

tion을 구성하는 데에 어려움이 있다. 이 경우에는 전문가의 조언을 exploration-exploitation 전략과 조합한 접근 방법으로 문제를 해결하는데, 다음 절에 이에 대한 구체적인 설명을 하기로 한다.

3.3 감독(Supervision)에 의한 ε_t-greedy

기본에이전트는 시간 t에서의 ϕ_t로 ε_t-greedy 전략을 사용하는 SARSA(0)를 학습 방법으로 한다. 2절에서 설명하였듯이, $c\in(0,1)$에 대하여 $1-c/n_t(x_t)$의 확률로 greedy한 행동 $a_t\in\arg\max_{a\in A}Q_t(x_t,a)$을 선택하고, $c/(|A|\,n_t(x_t))$의 확률로 $a\in A$인 행동 $a_t=a$를 선택한다(random한 선택을 의미한다). $n_t(x_t)$는 시간 스텝 t까지 상태 x_t로의 방문 횟수이다. ε_t-greedy는 모든 상태와 행동을 무한히 방문한다는 가정 하에 SARSA(0)의 수렴을 보장해 준다.

서브에이전트들, 그리고(또는) 전문가들에 의한 감독(supervision)에 의하여 SARSA(0)의 Q_t는 영향을 받게 된다. 따라서 $1-c/n_t(x_t)$의 확률로 선택된 greedy한 행동 a_t 역시 감독에 의하여 영향을 받게 된다. 하지만 $c/n_t(x_t)$의 확률로는 $|A|$개의 행동에 대하여 균등한 선택을 하게 되므로 결과적으로 "감독"은 ε_t-greedy 전략에 부분적으로만 영향을 미치게 된다. 서브에이전트들, 전문가들에 의한 감독을 잘 반영하기 위해 ε_t-greedy 전략을 다음과 같이 발전시킬 수 있다.

먼저, greedy한 행동 $a_t\in\arg\max_{a\in A}Q_t(x_t,a)$는 이전과 같이 $c\in(0,1)$에 대하여 $1-c/n_t(x_t)$의 확률로 선택된다. 하지만 $a\in A$라고 할 때 임의의 행동 $a_t=a$는 다음의 확률에 의해 선택된다.

$$\frac{c'}{n_t(x_t)}\max\left\{\theta(x_t,a;t_1,\ldots,t_m)\times\frac{\sum_{k=l}^{l}\rho_{s_k}^k(x_t,a)}{\sum_{a'\in A}\sum_{k=l}^{l}\rho_{s_k}^k(x_t,a')},\xi(x_t,a)\right\} \tag{12}$$

모든 $x\in X$, $a\in A$에 대해서 $\xi(x,a)\geq\delta>0$이고, $\sum_{a\in A}\xi(x,a)=1$이고, c'는 normalization을 위한 상수이다. 만일 $m=0$이면 $\theta(x_t,a;t_1,\cdots,t_m)$을 1로 정한다.

$t_x(i)$를 기본에이전트가 상태 x를 i번에 방문하는 시간 스텝이라고 하자. 또한 상태 x로의 i번째 방문에서 $a\in A$인 행동 a가 실행될 확률을 $\Pr\{a_t=a|x_t=x,t_x(i)=t\}$라고 하자. MDP M이 communicating하고 모든 $x\in X$, $a\in A$에 대해 다음이 성립한다고 가정하자:

$$\sum_{i=1}^{\infty}\Pr\{a_t=a|x_t=x,t_x(i)=t\}=\infty$$

그러면 반드시 모든 상태 $x\in X$가 무한히 방문되고, 각 상태에서 모든 행동 $a\in A$가 무한히 선택되어진다. 그런데 우리는 위에서 변형한 ε_t-greedy 전략으로부터 다음 식이 성립됨을 알 수 있다.

$$\Pr\{a_t=a|x_t=x,t_x(i)=t\}\geq\frac{c'\delta}{n_t(x_t)}$$

$\sum_{i=1}^{\infty}c''/i=\infty$, $c''\in(0,1)$이라는 점을 이용하면 우리는 변형된 ε_t-greedy 전략, 즉 감독된 ε_t-greedy 전략을 사용할 경우 모든 상태-행동 쌍이 무한히 반복되고 행하여지고, 결국엔 optimal한 행동으로 수렴함을 알 수 있다.

감독된 ε_t-greedy 전략에 의하면, 어떤 상태에 대한 방문 횟수가 적을 경우 에이전트는 서브에이전트, 그리고(또는) 전문가들의 감독된 정보를 따르는 경향이 크며, 반면에 방문 횟수가 많을 경우 greedy한 선택을 더 많이 하게 된다.

4. Policy Reuse 융합 기술

본 절에서는 에이전트가 어떤 문제(이를 task라고 표현할 것이다)에 대한 optimal policy를 학습을 통하여 얻은 뒤, 그것을 저장하여 다른 문제에 재사용하는, policy reuse 학습 기법에 대한 설명이 이루어질 것이다. 이 학습 방법을 사용하기 위해서는 먼저 1) exploration-exploitation 전략의 policy reuse 기법을 적용한 변형이 필요하고, 2) 각 task들의 유사성을 판별하여 policy를 재사용할 것인지 여부를 알려주는 유사성 함수(similarity function)에 대한 정의가 필요하며, 3) 유한한 수의 policy들을 저장할 policy 라이브러리를 유지해야한다[9]. **1)의 변형된 exploration 전략을 π-reuse exploration 전략, 2)의 유사성 함수를 적용한 알고리즘과 3)의 policy 라이브러리를 구성하고 유지하는 알고리즘을 각각 PRQ-학습 알고리즘, PLPR 알고리즘이라 한다.**

여기서는 policy reuse 기법의 핵심인 π-reuse exploration 전략에 대해서만 자세하게 설명하고, PRQ-학습 알고리즘과 PLPR 알고리즘에 대해서는 그 기본 개념만 제시하고 자세한 설명은 생략할 것이다.

과거에 학습을 통하여 얻어진 policy를 π_{past}라 하고, 이번에 학습하고자 하는 새로운 policy를 π_{new}라 하자. 그리고 에이전트는 ε_t-greedy 전략을 사용하는 SARSA(0)를 통한 학습을 하고 있다고 가정할 때 π-reuse 전략에서 $x\in X$인 상태 x에서 $a\in A$인 행동 a는 다음과 같이 선택된다.

$$a = \begin{cases} \pi_{past}(x) & \psi\text{의 확률} \\ \varepsilon_t - greedy(\pi_{new}(x)) & (1-\psi)\text{의 확률} \end{cases} \quad (13)$$

ε_t-greedy 전략에서 random한 행동과 greedy한 행동에 대한 선택 비율을 확률 ε_t를 사용하여 조절했던 것과 같이, **π-reuse 전략에서는 ψ를 사용하여 π_{past}에 의해서 행동을 선택할 것인지, 아니면 ε_t-greedy 전략에 의해서 행동을 선택할 것인지를 결정한다.** 즉 에이전트는 ψ의 확률로 $a=\pi_{past}(x)$, $(1-\psi)\varepsilon_t$의 확률로 random 행동을, $(1-\psi)(1-\varepsilon_t)$의 확률로 greedy한 행동을 선택하는 것이다($\varepsilon_t = c/n_t(x_t)$라고 하자). 확률 ψ의 값을 학습이 진행됨에 따라 점점 작게 주면 학습 초기에는 과거의 policy에 의존한 학습을 하게 되고 학습이 진행될수록 독립적인 학습이 가능하게 된다.

π-reuse 전략은 학습 초기에 π_{past}를 통해 학습 방향을 제시해 주어 학습에 영향을 미친다. 에이전트가 학습을 통해 해결하고자 하는 문제 상황을 task라 하고 Ω로 표현하자. 과거의 task를 Ω_{past}, 현재의 task를 Ω_{new}라고 할 때, Ω_{past}와 Ω_{new}의 유사도가 높을수록 π- reuse 전략을 사용한 학습의 효율(수렴 속도)이 향상되고, 지나치게 낮을 경우에는 π-reuse 전략을 사용하지 않을 때보다도 좋지 않은 결과를 보이게 된다. **PRQ-학습 알고리즘은 task들 간의 유사도를 측정하는 유사성 함수(similarity function)를 사용하여 task에 따라 π-reuse 전략의 사용 여부와 policy 라이브러리에서 어떤 policy를 재사용할 것인지를 결정함으로써 위의 문제를 해결한다.**

PLPR 알고리즘은 policy 라이브러리를 구성하고 새로운 policy를 라이브러리에 추가할 것인지, 추가할 경우 기존의 라이브러리에서 어떤 policy를 삭제할 것인지에 대해서 역시 유사성 함수를 기반으로 결정한다.

policy reuse 기법은 학습 초기에 에이전트에게 과거의 policy를 통한 힌트를 제시해 주며, 이는 2절에서 소개되었던 다중 학습에 서브에이전트와 전문가 어떤 형태로든 조합이 가능한 학습 기법이다.

5. 결 론

지금까지 최근 활발한 연구가 진행되고 있는 두 가지 강화 학습에 대해서 알아보았다. 하나는 여러 학습자들의 독립적인 학습과 전문가들이 감독(supervision)의 형태로 강화 학습과 조합된 새로운 학습 기법이었고, 다른 하나는 학습의 결과물을 저장하는 라이브러리를 구성하여 그것을 새로운 학습에 재사용하는 학습 기법이었다.

Policy reuse 기법은, 동일한 조건에서 목적지만 바뀌는 navigation 문제와 같은 상황에 좋은 성능을 낼 것으로 보인다. 그리고 policy reuse 기법이 서브에이전트나 전문가의 형태로 다른 학습과 조합하여 보다 좋은 성능을 내는 것 역시 가능하다.

먼저 소개한 학습 기법의 가장 큰 장점은 서브에이전트들이 어떤 종류의 학습을 하더라도, 전문가들이 어떤 시스템을 기반으로 하더라도 별다른 제한 없이 조합이 가능하다는 점이다. 이 학습 기법을 통하여 다양한 학습 기법의 조합을 통한 성능 향상이 용이해 질 것으로 보인다.

참고문헌

[1] R. Sutton and A. Barto, Reinforcement Learning. MIT Press, 2000.

[2] D. P. Bertsekas and J. N. Tsitsiklis, Neuro Dynamic Programming. Athena Scientific, 1996.

[3] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[4] S. Russel and A. L. Zimdars, "Q-decomposition for reinforcement learning agents," in Proc. of the 20th Int. Conf. on Machine Learning, 2003, pp. 278-287.

[5] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," Machine Learning, Vol. 16, pp. 185-202, 1994.

[6] S. Singh, T. jaakkola, M. Littman, and C. Szepesvari, "Convergence results for single-step on-policy reinforcement learning algorithms," Machine Learning, Vol. 38, pp. 287-308, 2000.

[7] A. Y. Ng, D. Harada, and S. Russel. "Policy invariance under reward transformations: theory and application to reward shaping," in Proc. of the 16th Int. Conf. on Machine Learning, 1999, pp. 278-287.

[8] H. S. Chang, "Reinforcement Learning with Supervision by Combining Multiple Learnings and Expert Advices," in Proc. of the 2006 American Control Conference, June, 2006, pp. 4159-4164.

[9] F. Fernandez and M. Veloso, "Probabilistic Policy Reuse in a Reinforcement Learning Agent," In The **Fifth International Joint Conference on Autonomous Agents and Multia-**

gent Systems, May, 2006.

[10] A. G. Barto, "Reinforcement Learning" in Handbook of Learning and Approximate Dynamic Programming, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch (eds.), pp. 804-809, Wiley-IEEE Press, Piscataway, NJ, 2004.

[11] M. N. ahmadabadi and M. Asadpour, "Expertness based cooperative Q-learning," IEEE Trans. on Systems, Man, and Cybernetics, part B. Vol. 32, No. 1, pp. 66-76, 2002.

[12] A. G. Barto and M. T. Rosentein, "Supervised Actor-Critic Reinforcement Learning," in Handbook of Learning and Approximate Dynamic Programming, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch (eds.), pp. 359-380, Wiley-IEEE Press, Piscataway, NJ, 2004.

[13] M. Rosentein and A. G. Barto, "Reinforcement learning with supervision by a stable controller," in Proc. of the American Control Conf., 2004, pp. 4517-4522.

[14] K. Driessens and S. Dzeroski, "Integrating experimental and guidance in relational reinforcement learning," in Proc. of the 19th Int. Conf. on Machine Learning, 2002, pp. 115-112.

김 성 완

2006년 서강대학교 컴퓨터공학과(학사)
2006년~현재 서강대학교 컴퓨터공학과 석사과정
관심분야 : Reinforcement Learning, Computational Intelligence
E-mail : inaina21@sogang.ac.kr

장 형 수

1994년 미국 Purdue University, 전기 및 컴퓨터 공학과 졸업
1996~2001 동대학원 석사, 박사
2001년 9월~2003년 6월 Institute of Systems Research, University of Maryland, College Park, Research Associate
2002년 6월~2003년 2월 고려대학교 정보통신기술공동연구소 연구교수
2003년 3월~현재 서강대학교 공과대학 컴퓨터공학과 조교수
E-mail : hschang@sogang.ac.kr