효율적인 멀티 에이전트 강화 학습을 위한 나이브 베이지안 기반 상대 정책 모델

A Naive Bayesian-based Model of the Opponent's Policy for Efficient Multiagent Reinforcement Learning

권 기 덕*
Ki-Duk Kwon

요 약

멀티 에이전트 강화학습에서 중요한 이슈 중의 하나는 자신의 성능에 영향을 미칠 수 있는 다른 에이전트들이 존재하는 동적 환경에서 어떻게 최적의 행동 정책을 학습하느냐 하는 것이다. 멀티 에이전트 강화 학습을 위한 기존 연구들은 대부분 단일 에이전트 강화 학습기법들을 큰 변화 없이 그대로 적용하거나 비록 다른 에이전트에 관한 별도의 모델을 이용하더라도 현실적이지 못한 가정들을 요구한다. 본 논문에서는 상대 에이전트에 대한 나이브 베이지안 기반의 행동 정책 모델을 소개한 뒤, 이것을 이용한 강화 학습 방법을 설명한다. 본 논문에서 제안하는 멀티 에이전트 강화학습 방법은 기존의 멀티 에이전트 강화 학습 연구들과는 달리 상대 에이전트의 Q 평가 함수 모델이 아니라 나이브 베이지안 기반의 행동 정책 모델을 학습한다. 또한, 표현력은 풍부하나 학습에 시간과 노력이 많이 요구되는 유한 상태 오토마타나 마코프 체인과 같은 행동 정책 모델들에 비해 비교적 간단한 형태의 행동 정책 모델을 이용함으로써 학습의 효율성을 높였다. 본 논문에서는 대표적인 적대적 멀티 에이전트 환경인 고양이와 쥐게임을 소개한 뒤, 이 게임을 테스트 베드 삼아 실험들을 전개함으로써 제안하는 나이브 베이지안 기반의 정책 모델의 효과를 분석해본다.

Abstract

An important issue in Multiagent reinforcement learning is how an agent should learn its optimal policy in a dynamic environment where there exist other agents able to influence its own performance. Most previous works for Multiagent reinforcement learning tend to apply single-agent reinforcement learning techniques without any extensions or require some unrealistic assumptions even though they use explicit models of other agents. In this paper, a Naive Bayesian based policy model of the opponent agent is introduced and then the Multiagent reinforcement learning method using this model is explained. Unlike previous works, the proposed Multiagent reinforcement learning method utilizes the Naive Bayesian based policy model, not the Q function model of the opponent agent. Moreover, this learning method can improve learning efficiency by using a simpler one than other richer but time-consuming policy models such as Finite State Machines(FSM) and Markov chains. In this paper, the Cat and Mouse game is introduced as an adversarial Multiagent environment. And then effectiveness of the proposed Naive Bayesian based policy model is analyzed through experiments using this game as test-bed.

☞ keyword : Multiagent, reinforcement learning, Naive Bayesian

1. 서 론

환경의 변화에 영향을 미칠 수 있는 다수의 에이전트가 공존하는 멀티 에이전트 환경에서 중요한 문제점 중의 하나는 한 에이전트가 시행착오적 상호 작용을 통해 어떻게 자신의 최적 행동 정책을 학습해 나갈 수 있느냐는 것이다. 멀티 에이전트 환경에서의 에이전트들은 자율적으로 자신의 행동 정책을 학습할 수 있으며 시간이 경과함에 따라서 상태 전이 함수가 변할 수 있다. 이런 환경적 특성을 가진 멀티 에이전트 환경에서 자신의 행동 정책

* 정 회 원 : 경기대학교 전자계산학과 박사과정
kdkwon@kyonggi..ac.kr
[2008/07/31 투고 - 2008/08/05 심사 - 2008/09/03 심사완료]

을 좀 더 효율적으로 학습하기 위해서는 상대에 대한 모델을 만들어 멀티 에이전트 강화 학습을 하는 방법들에 대한 연구들이 진행되어 왔다. 그동안 다른 에이전트의 존재를 고려한 멀티 에이전트 강화 학습의 대표적인 연구 중의 하나는 두 명의 제로-합 게임을 위한 Littman의 Minimax-Q 학습 방법[1]으로 자신뿐만 아니라 상대 에이전트도 자신과 같은 Q 학습에 의해 행동을 선택한다고 가정하고 자신이 받은 보상 값을 토대로 두 에이전트의 연합 행동(joint action)에 대한 Q-함수를 학습한다. 그러나 이 학습 방법은 상대 에이전트의 Q 학습 뿐 만 아니라 보상 값까지도 알아야 하는 비현실적인 가정을 하고 있다. Hu와 Wellman이 제안한 Nash-Q 학습 방법[1]은 제로-합 게임에 적용 가능한 Minimax-Q 학습 방법을 협력 에이전트들이 존재하는 일반-합 게임에 적용 가능 하도록 확장한 학습 방법이다. 이 학습 방법은 자신 뿐 만 아니라 상대의 Q 함수도 저장하고 있어야 하기 때문에 많은 저장 공간을 필요로 하며 따라서 학습 수렴 속도가 느리다는 것과 Nash 평형을 찾아야 하는 문제점을 가지고 있다. Tesarou가 제안한 Hyper-Q 학습 방법[14]은 관찰된 행동을 바탕으로 학습하는 것이 아니라 상대의 정책을 바탕으로 학습한다. 따라서 현실적이지 못한 문제점을 가지고 있다. 이와 같은 문제점들을 해결하기 위한 저자의 선행 연구로 과거에 관찰된 상대의 행동들을 기초로 상대의 행동 정책 모델을 학습하여 이 모델을 바탕으로 단순 멀티 에이전트 강화 학습 방법을 사용하였다. 그러나 선행 연구의 경우 상대 행동 정책 모델을 이용함에 있어 상대의 행동 선택이 확률적이지 않기 때문에 효율적이지 못한 문제점을 가지고 있다. 상대 행동 정책 모델은 0과 1 사이의 값을 가지는데 이 값은 실제 수행한 행동에 대한 효과를 조절하기 위한 θ 값에 의해 조정된다. 본 논문에서는 나이브 베이지안 (Naive Bayesian)을 이용하여 비교적 가벼운 확률 모델로 상대 에이전트의 행동 정책 모델을 생성하여 이 모델을 적용하여 자신의 최적 정책을 학습하는 강화 학습방법을 제시한다. 본 논문에서 제안하는 멀티 에이전트 강화 학습방법은 제로-합 확률 게임으로 두 명의 에이전트로 구성된 적대적 멀티 에이전트 환경을 가정하며, 두 에이전트는 동시에 행동을 수행함으로써 자신의 행동을 결정하기 전에 미리 상대 에이전트의 행동을 알 수는 없으나 일단 동시에 행동을 수행하고 나면 상대 에이전트가 수행한 행동을 관찰할 수 있다. 하지만 두 에이전트 간에는 행동 결정에 영향을 미치는 어떠한 통신도 가능하지 않다고 가정한다. Q 학습 알고리즘을 확장한 이 멀티 에이전트 강화학습 방법은 상대 모델을 이용하는 기존의 멀티 에이전트 강화 학습 연구들에서 주로 시도되었던 상대 에이전트의 Q 평가 함수 모델 대신 상대 에이전트의 행동 정책 모델 수립에 나이브 베이지안을 이용함으로써 학습의 효율성을 높였다. 본 논문의 구성은 상대 모델에 대한 기본 개념을 살펴보고 멀티 에이전트 강화 학습 방법에 상대 모델의 일반화 방법인 나이브 베이지안을 적용해 본다. 그리고 대표적인 적대적 멀티 에이전트 환경인 고양이와 쥐게임을 소개하고 이 게임을 테스트 베드 삼아 수행한 비교 실험 결과들을 설명함으로써 본 논문에서 제안하는 나이브 베이지안 기반 상대 정책 모델 기반의 멀티 에이전트 강화 학습의 효과를 분석해 본다.

2. 관련 연구

2.1 멀티에이전트 강화 학습

단일 에이전트 환경에서 에이전트가 환경과의 상호 작용을 통해 자신의 행동에 대한 보상 값을 바탕으로 최적의 행동 정책을 수립해 나가는 과정은 하나의 마코프 결정 문제(Markov Decision Problem, MDP)로 정의할 수 있다. 그러나 다수의 에이전트들로 구성되어 에이전트들 간에 서로 상호 작용하는 멀티 에이전트 환경은 하나의 MDP로 표현 할 수 없다. 그러나 MDP를 행동을 결정하는 에이전트가 다수 참여하는 멀티 에이전트 환경으로 일반화하는 확률 게임으로 정의할 수 있다[1]. 따라

서 확률 게임에 참여하는 각 에이전트들은 자신의 행동에 대한 보상 값과 게임 상태 전이에 영향을 주는 다른 에이전트의 존재를 고려하여 자신의 최적 행동 정책을 학습하는 것이다.

멀티 에이전트 환경에서 상대 정책 모델을 이용한 기존의 멀티 에이전트 강화 학습들은 주로 다른 에이전트들에 대한 고려를 어떻게 자신의 최적 행동 정책 학습에 반영했는지에 대한 해법을 제시한 것으로 볼 수 있다. 특히 멀티 에이전트 강화 학습을 위한 다른 에이전트의 가치 함수(value function)나 정책(policy)에 대한 상대 모델을 명시적으로 이용하는 경우와 이용하지 않는 경우로 기존 연구들을 분류할 수 있는데, 이용하지 않는 경우라도 사전에 다른 에이전트들의 정책이나 행동 양식에 대한 특별한 가정(assumption)을 기초로 학습을 진행한다. 따라서 이 가정이 상대에 대한 모델이 될 수 있다.

(표 1) 상대 모델 기반 멀티 에이전트 강화학습

	Minimax-Q	Nash-Q	Hyper-Q
대상	상태 평가 함수	Q 함수	혼합 전략
목적	지식 확장	행동 정책 학습	행동 정책 학습
가정	Q 함수, 보상 값 알아야 한다.	Q 함수 저장	상대 전략 저장

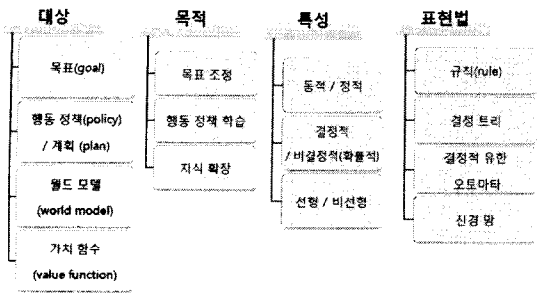
상대 모델을 이용하여 멀티 에이전트 강화 학습을 하는 경우는 Minimax-Q[2]와 Nash-Q[1], Hyper-Q[14] 등이 있다.

Littman이 제안한 Minimax-Q 학습은 두 명의 에이전트로 구성된 제로-합 게임을 위한 강화학습 방법이다[2]. Minimax-Q 학습에서 학습 에이전트는 상대가 선택할 수 있는 최악의 행동을 감안하여 학습 에이전트가 얻을 수 있는 기대 보상 값을 최대화할 수 있는 최적 정책을 학습한다. Minimax-Q 학습은 학습 에이전트가 받은 보상 값을 토대로 두 에이전트의 연합 행동(joint action)에 대한 Q-함수 $Q_1(s, (a_1, a_2))$을 학습하며 상대 에이전트의 학습 에이전트와 같은 Q 학습 방법에 의한 행동을 모델로 사용한다. 그러나 이 학습 방법은 상대 에이전트가 언제나 최악의 행동선택을 할 것이라는 가정을 토대로 하고 있으며, 두 명의 제로-합 게임에만 적용 가능하다는 제한점이 있다.

Hu와 Wellman이 제안한 Nash-Q 학습 방법[6]은 제로-합 게임에 적용 가능한 Minimax-Q 학습 방법을 협력 에이전트들이 존재하는 일반-합 게임에 적용 가능 하도록 확장한 학습 방법이다. Nash-Q 학습은 Minimax-Q 학습 알고리즘과는 달리 일반-합 게임에 적용 가능하다는 장점은 있으나, 학습 에이전트 자신뿐만 아니라 다른 에이전트들의 Q 함수 값까지 유지하기 위해 다른 에이전트들의 보상 값과 학습 파라미터들(learning parameters)까지 알아야 한다는 제한점이 있다. Tesarou는 Hyper-Q 방법[14]을 제안하였는데, 이 방법은 학습 에이전트의 행동 정책 수립을 위한 강화 학습과정에서 상대 에이전트와 학습 에이전트의 연합 행동에 의한 Q 학습이 아니라 상대와 학습 에이전트의 정책에 의한 혼합 전략(Mixed Strategy)을 이용하여 학습한다. 상대의 정책은 나이브 베이지안 방법을 사용하며 관찰된 행동에 유틸리티 값을 주었다.

멀티 에이전트 환경에서 자신의 최적 정책을 수립하기 위해서는 상대 에이전트를 고려하여 학습이 이루어져야 한다. 따라서 어떻게 상대 에이전트에 대한 정보를 알아야 하는가는 중요한 문제점 중 하나이다. 이를 해결하기 위한 방법으로 상대 에이전트를 모델링 한다.

그림 1은 상대 모델의 분류 체계도이다. 분류의 기준은 크게 4가지로 모델링의 대상과 적용, 특성, 그리고 표현법에 따라 분류한다. 모델링의 대상은 크게 4가지로 나눠 볼 수 있으며, 적용 목적에 따라 달리 표현된다. 첫째는 상대의 목표(goal)를 대상으로 모델링 하는 것으로 상대 에이전트 뿐 만 아니라 다른 에이전트의 목표를 모델링한다. 팀으로 구성되는 멀티 에이전트 환경에서 자신의 목표와 다른 목표를 가지는 에이전트들을 모델링함으로써 자신의 목표를 팀의 목표를 달성하기 위해 조정



(그림 1) 상대 모델 분류 체계도

할 수 있다. 상대 에이전트의 목표를 모델링하는 것은 자신과 같은 목표를 상대도 똑같이 가진다는 가정 하에 이루어진다. 두 번째는 월드 모델(world model)을 모델링하는 것이다. 월드 모델을 모델링 한다는 것은 상대 에이전트를 환경의 일부분으로 가정하여 상대 에이전트에 의한 월드 모델의 변화를 모델링 하는 것으로 자신의 도메인 지식을 넓힌다. 세 번째로 멀티 에이전트 분야에서 가장 많이 이용하는 방법인 상대의 정책 또는 계획(plan)을 모델링 하는 것으로 상대의 행동 정책은 학습을 통해 수립된다. 마지막으로 상대의 가치 함수를 모델링 한다. 상대의 가치 함수는 에이전트가 현재 상태에 있는 것이 목적 상태에 도달하는데 어느 정도 도움이 되는지 또는 에이전트가 현재 상태에서 특정 행동을 수행하는 것이 어느 정도 가치가 있는가를 평가한다. 이 평가는 자신의 행동 정책 수립을 위한 학습에 이용한다.

상대 모델의 특성은 다음과 같이 나뉘어 볼 수 있다.

선형/ 비선형: 모델은 일반적으로 변수들과 변수들의 행동 오퍼레이터 등으로 구성된다. 만일 모든 모델의 행동 오퍼레이터들이 선형적인 결과를 가지는 모델이라면 선형의 특징을 가지게 되고 그렇지 않다면 비선형적이다 라고 말 할 수 있다.

결정적 / 확률적 : 결정적 모델은 변수들의 조합으로 이전 상태들의 집합을 구성하거나 모델의 파라미터에 의해 결정되는 상태 변수들의 집합 중의 하나이다. 그래서 결정적 모델은 주어진 초기 조건들의 집합에 의해 결정 사항이 항상 같은 방법으로 수행된다. 반대로 확률적 모델은 현재는 랜덤하게 결정 사항을 수행하며 상태 변수들은 유일한 값들로 표현하지 않고 확률 분포에 의해 표현한다.

동적 / 정적 : 정적 모델은 모델을 구성하는데 있어서 시간을 고려하지 않는다. 동적 모델은 시간을 고려하며, 모델의 표현은 주로 미분식이나 차방정식을 이용한다.

멀티 에이전트 환경에서 상대 모델의 표현 방식은 결정 트리, DFA, 신경 망, 규칙 기반 등이 있다.

결정 트리를 사용하여 상대 모델을 학습한 연구로 의사 결정 이론에서 결정 트리는 의사 결정을 그래프 형태로 표현하며, 의사 결정들의 가능한 결과들은 목표(goal)를 달성하기 위한 계획(Plan)을 생성하는데 이용한다. 결정 트리는 예측 모델이다[6]. 이는 어떤 아이템의 목표(target) 값에 대해 결론짓기 위해 아이템에 대한 관찰을 시도한다. Ramon과 Jacobs 등은 바둑 게임에서 이미 만들어진 전문가에 의한 학습된 플레이를 모델링의 대상으로 결정 트리를 표현하여 상대의 행동을 예측하였다[9]. 학습된 플레이의 분석은 일반적인 플레이가 아닌 특이한 플레이를 분석하여 잘 알지 못하는 상대의 행동을 예측한다. 그러나 전문가의 학습된 모델이 포함하지 않는 행동에 대한 예측이 발생할 수 있으며, 정확한 상대 행동의 예측을 하기 위해 많은 양의 모델을 필요로 한다.

Carmel과 Markovitch는 상대의 정책을 모델링의 대상으로 상대의 행동 정책을 학습하기 위해 DFA (Deterministic Finite Automata)로 모델을 표현한다고 가정한다[10]. 이 가설에 의해 Carmel과 Markovitch가 제안한 US-L* 알고리즘은 에이전트의 행동 기록과 일치하는 DFA를 학습한다. 결과로서 상태-행동 쌍으로 표현되는 부분 집합 간의 모델의 상관관계를 알 수 있다. Carmel과 Markovitch는 2명의 반복 게임(two-player repeated game)인 가위 바위 보게임에서 과거 행위에 기반을 둔 상대 전략을 DFA로 표현하여 모델링하고 미래의 행위를 모델을 사용하

여 예측한다[10]. 그러나 게임의 복잡도가 매우 낮은 환경에서 실험하였고 모델은 갱신되지 않고 한 번 만들어지면 고정된 정책 모델로 인정하여 상대의 행동을 DFA를 이용하여 예측하였다.

Davidson은 포커 게임에서 상대 행동인 fold, call, raise 등을 예측하기 위해 실행 가능한 행동들을 모델링의 대상으로 신경망을 이용하여 모델을 수립하였다[11]. 정확한 예측을 위해 다른 많은 에이전트들과의 게임을 통해 축적된 게임 맥락(context)을 많이 가지고 있어야 한다. 신경망의 입력 값은 축적된 게임 맥락, 출력 값은 예측되는 상대의 행동이다. 이 예측 행동에 대한 정확성을 높이기 위해 가중치 값을 조정한다. 그러나 신경 망 모델은 학습하는데 걸리는 시간이 오래 걸리고 유효한 예측을 하려면 많은 훈련 예제를 필요로 하고 훈련 예제들은 상태 공간에 골고루 분포되어 있어야 한다. 또한 신경망은 블랙박스로서 예측하는 과정을 설명할 수 없다.

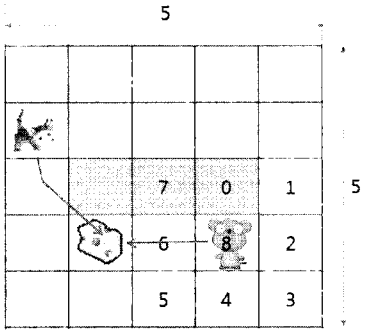
미래의 행동을 예측하기 위한 가장 쉬운 방법은 주어진 현재 상태에서 미리 정의된 규칙(rule) 집합을 이용하는 것이다[5]. 이 방법은 모든 가능한 상태에 가능한 행동들의 확률 분포가 주어지는 경우에 바탕을 둔(case-based) 결정 사항들로 구성된다. 이 접근 방법을 사용하는 경우 훈련에 필요한 기간이 필요 없고 이미 도메인에 대한 자세한 정보가 정형화된 규칙 집합으로 구성되어 있기 때문에 도메인 의존적이어서 이 방법에 의한 모델링의 경우 모델링 초기에 유용하게 이용될 수 있는 방법이다.

Veloso는 로봇 축구에서 자율적인 로봇 행위의 인식을 위해 HMM(Hidden Markov Model)을 이용하였다[13]. HMM은 분산된(discrete) 상태들의 집합으로 표현한다. 상태 전이는 에이전트들의 연합 행동에 의한 확률 분포에 의해 진행된다. 상태 정보는 그 시각에 직접 관찰 할 수 없고 상태 집합을 구성하는 관찰 변수들을 이용한다. 그러나 HMM은 다수 에이전트에 대한 행동을 인식하기에는 모델이 매우 복잡하다.

베이지안을 이용한 모델 학습 방법의 연구는 주로 베이지안 망에 의한 학습 연구가 주를 이룬다. 대표적인 베이지안 망 학습 연구로 Price와 Boutilier는 학습자(learner) 에이전트가 고정된 정책을 사용하는 멘토(mentor) 에이전트의 정책 모델을 학습하는데 확률적 접근 방법인 베이지안 망 방법을 이용하였다[12]. 이는 고정된 정책을 대상으로 모델링하기 때문에 반복적 행동 등을 유발할 수 있으며, 이는 학습 에이전트의 행동 정책 수립을 위한 충분한 모델을 제시하지 못한다. 또한 Boutilier는 확률 게임에서 멀티 에이전트가 서로 협조(Coordination)하기 위해 팀 구성원을 대상으로 구성원의 행동들을 모델링하기 위해 베이지안 방법을 이용하여 모델링한다[12]. 이는 서로 다른 목적(goal)을 가지는 에이전트들의 연구로서 본 논문에서 제시하는 적대적 환경에서의 멀티 에이전트 환경에서는 적용하기 어렵다.

3. 예제

고양이와 쥐게임(Cat and Mouse game)은 멀티 에이전트 연구에 많이 이용되어 온 추적 게임(pursuit game)의 한 변형이다. 이 게임은 전형적인 제로-합 확률 게임으로서, 적대관계인 두 명의 에이전트가 서로 생사를 걸고 경쟁을 벌이는 멀티 에이전트 환경이다. 이 게임에서 고양이와 쥐는 서로 상반된 목표를 가지고 있다. 고양이는 쥐를 잡아 점수를



(그림 2) 고양이와 쥐게임 문제
(▨: 장애물, 숫자 : 수행 행동)

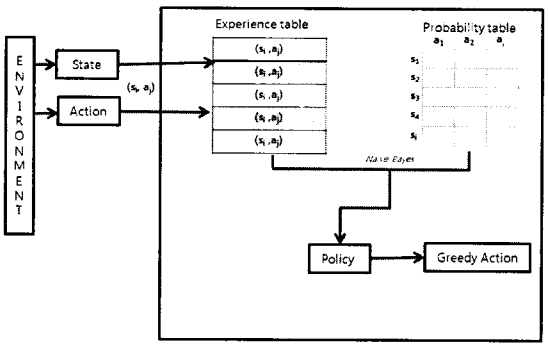
획득하는 것이 목표이며, 반면에 쥐는 고양이에게 잡히지 않으면서 많은 치즈를 먹어 점수를 높이는 것이 목표이다. 임의의 위치에서 시작하여 도망가는 쥐와 쫓아가는 고양이가 같은 셀에 위치하게 되면 고양이가 쥐를 잡은 것으로 간주하고 게임이 종료된다.

쥐와 고양이는 그림 2와 같이 벽과 장애물들이 존재하는 5 x 5의 2차원 격자월드(grid world)에서 동시에 동, 서, 남, 북, 북동, 남동, 남서, 북서, 제자리 등 아홉 방향에 위치한 인접 셀(cell) 중 하나로 이동(a0 ~ a8)할 수 있다. 단, 벽이나 장애물이 놓여있는 셀로의 이동은 허용되지 않다. 따라서 그림 2에서 쥐의 행동은 <a1, a2, a3, a4, a5, a6, a8>이다. 장애물인 0과 7의 위치로는 쥐의 이동이 이루어지지 않는다. 고양이의 행동은 장애물이 있는 a3의 행동과 벽인 <a5, a6, a7>의 행동은 발생하지 않는다. 게임의 상태(s)는 고양이의 위치좌표, 쥐의 위치좌표, 치즈의 위치좌표의 조합으로 표현한다. 그림 2의 환경에서 쥐의 최적 행동은 우측으로 이동(a6)하는 것으로 치즈와의 거리를 줄이는 방향으로 이동한다. 반면에 고양이는 쥐와의 거리를 줄이기 위해 최적 행동(a4)을 한다. 본 논문에서는 쥐가 치즈를 많이 먹고 고양이를 피해 다니기 위해 고양이의 상대 모델을 이용한 멀티 에이전트 강화 학습을 이용한다. 따라서 상태집합의 크기 $|S|$는 (5 x 5)3 = 15625 이고, 가능한 연합 행동들의 수 $|\vec{A}|$는 9 x 9 = 81 이므로, 전체 상태공간의 크기는 15625 x 81 = 1265625이다. 고양이와 쥐게임에서 상태전이는 언제나 현재 상태와 에이전트들이 수행한 연합 행동에 따라 일정한 하나의 후속 상태가 정해진다. 따라서 고양이와 쥐게임에서 상태전이는 하나의 결정적 함수(deterministic function)로 표현될 수 있다. 그리고 쥐는 치즈를 먹었을 때, 고양이는 쥐를 잡았을 때 각각 일정한 양의 보상 값을 받는 것으로 가정한다.

4. 나이브 베이지안 상대 모델 학습

학습 에이전트는 자신의 행동 정책 수립을 위한 효율적인 멀티 에이전트 강화 학습을 위해 상대 정책 모델을 이용한다. 상대 모델을 이용하는 기존의 멀티 에이전트 강화 학습 연구들에서 주로 시도되었던 상대 에이전트의 Q 평가 함수 모델 대신 상대 에이전트의 행동 정책 모델을 학습하며, 표현력은 풍부하나 학습에 시간과 노력이 많이 요구되는 확률 모델인 베이지안 망(bayesian network)이나 영향 다이어그램(influence diagram)과 같은 행동 정책 모델들에 비해 비교적 간단한 형태의 확률적인 나이브 베이지안 방법을 이용하여 상대 행동 정책 모델을 학습한다.

그림 3에 의하면 상태는 쥐와 고양이, 그리고 치즈의 격자 공간상의 x, y 좌표이다. 이 상태 정보를 바탕으로 몇 번의 학습이 끝나면 상대의 행동에 대한 모델이 만들어지게 된다. 이 때 수립되는 모델은 상대 에이전트의 행동을 관찰함으로써 만들어지며 각각의 행동이 각 상태에서 어느 정도 수행되는지의 확률 테이블을 만든다. 이 확률 테이블이 경험(Experience, E) 테이블이며, 이 E 테이블 값에 각각의 상태에서 각 행동이 나올 확률 값을 곱하여 확률(Probability, P) 테이블이 만들어 진다. 이 P 테이블의 값과 E 테이블의 값을 고려하여 나이브 베이지안에 의한 정책(Policy)을 만들게 되며 정책에 의해 지금까지 실행한 행동들의 집합을 기준으로



(그림 3) 상대 정책 모델 학습 과정

다음에 어떤 행동을 실행하면 유추해서 이런 행동을 할 것이라고 예측한다. 이와 같은 과정을 반복 수행함으로써 모델을 만들게 된다.

NaïveBayesianLearn (observation_list)
/* observation_list = {(s_0, a_0), (s_1, a_1), ... (s_i, a_j)} */
/* 여기서 s_i는 상태, a_j는 상대 에이전트의 행동을 나타낸다. */

/* 각 행동 a_j에 대한 사전 확률(priori probability)을 계산한다 */
For each action a_j in observation_list
$\hat{P}(a_j) \leftarrow$ Estimate($P(a_j)$);

/* 각 상태s_i 마다 실행될 수 있는 각 행동들의 확률 값을 구한다. */
For each state s_i in observation_list

$$\hat{P}(a_i|s_j) \leftarrow \frac{P(a_j)P(s_i|a_j)}{P(S_i)};$$

/* 생성된 모델을 바탕으로 다음에 같은 상태를 만나면 모델을 갱신한다. */
Classify New (state, action) pairs ((s_i,a_j))

$$a_{NB} = \underset{a_j \in A}{\arg\max} \prod_{s_i \in S} \hat{P}(a_i|s_j)$$

(그림 4) 나이브 베이지안 학습 알고리즘

그림 4에서 $\hat{P}(a_j)$은 각각의 행동에 대한 확률 값을 평가하는 것이다. 그리고 각 상태마다 각 행동에 대한 확률 값을 평가하기 위한 $\hat{P}(a_i|s_j)$을 수행한다. 나이브 베이지안 학습에서 s_i를 i번째 상태라고 하고 a_j를 해당 행동이라고 하면 예측치 a_{NB}를 구하는 기본적인 나이브 베이지안 학습 방법은 그림 4의 $a_{NB} = \underset{a_j \in A}{\arg\max} \prod_{s_i} \hat{P}(a_i|s_j)$과 같이 갱신된다.

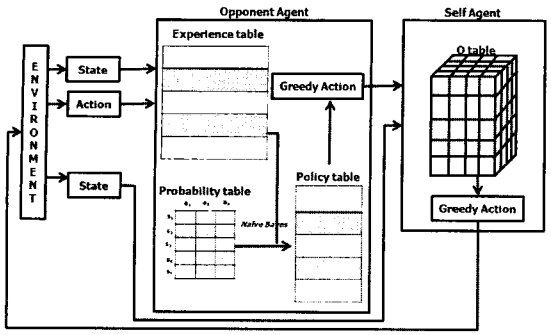
그림 2의 환경을 예로 들면, 현재 상태가 s_1=<3,3,0,1,1,3> 이라면 이 때 고양이의 실제 행동이 a_4이다. 여기서 상태 <3,3,0,1,1,3>는 격자 상 쥐의 x좌표가 3, 쥐의 y좌표가 3, 고양이의 x좌표가 0, 고양이의 y좌표가 1, 그리고 치즈의 x, y 좌표가 각각 1, 3 임을 말한다. 이런 상태가 들어오면 그림 1의 E 테이블에 현재 상태와 그 때의 실제 행동을 저장하게 되며, 만약 P 테이블에 현재 상태에 대한 예측 행동 정보가 있다면 이 2개 테이블의 값들을 고려하여 식 2에 의해 행동 확률이 가장 높은 상태 정보와 행동을 정책으로 간주하여 이 정책에 의해 고양이는 현재 상태에서 어떤 행동을 수행한다는 모델을 쥐는 가지게 된다.

만약 현재 상태가 이전에 경험하지 못했던 새로운 상태 정보라면 E 테이블에 현재 상태의 정보와 관찰된 행동 정보를 저장하게 되고 고양이는 고양이의 정책에 의해 행동을 수행하게 되며 P 테이블에 현재 상태와 행동에 대한 확률 값을 저장하게 된다.

5. 상대 모델의 적용

본 논문에서 가정하는 적대적 멀티 에이전트 환경은 두 명의 제로-합 확률 게임으로서, 적대관계의 두 에이전트가 동시에 각각 자신의 행동을 수행하는 동기화된 환경이다. 그리고 두 에이전트는 서로 관찰을 통해 상대방이 수행하는 행동을 알 수 있으나, 서로 간에는 어떤 통신도 없다고 가정한다. 두 명의 제로-합 확률 게임(two-player zero-sum SG)은 튜플 <S, $\vec{A}$, T, $\vec{R}$>로 정의한다. $S = \{s \mid s = (s_{self}, s_{opponent})\}$ 는 게임 상태들의 유한 집합, $\vec{A} = \{(a_{self}, a_{opponent}) \mid a_{self} \in A_{self}, a_{opponent} \in A_{opponent}\}$는 연합 행동(joint action)들의 집합, 이때 각각 A_{self} 와 $A_{opponent}$ 는 에이전트 Self와 에이전트 Opponent가 선택할 수 있는 행동들의 집합, T: S X $\vec{A}$ -> Π(S)는 에이전트 Self와 에이전트 Opponent의 연합 행동에 따라 다음 상태를 결정하는 상태전이함수, $\vec{R} = R_{self}, R_{opponent}$ 는 에이전트 Self와 에이전트 Opponent의 보상함수, 이때 $R_{self} : S \times \vec{A} \rightarrow R$ 는 에이전트 Self의 보상 값을, $R_{opponent} : S \times \vec{A} \rightarrow R$ 는 에이전트 Opponent의 보상 값을 결정하는 보상함수를 말한다.

그림 5는 전체적인 나이브 베이지안 방법을 이용한 멀티 에이전트 Q 학습의 구조를 나타낸다. 상태가 주어지고 상대 행동이 관찰되면 경험 테이



(그림 5) 멀티 에이전트 강화 학습 과정

블(Experience table)에 상태와 행동이 누적된다. 그리고 상태와 각각의 행동에 대해 확률 테이블(Probability table)이 만들어 진다. 초기에는 확률 테이블이 어떠한 확률 값도 가지지 않지만 시간이 흐름에 따라 확률 테이블에는 하나의 상태에서 더 많이 관찰된 행동의 확률 값이 높아진다. 확률 테이블과 경험 테이블을 바탕으로 나이브 베이지안 방법을 이용하여 정책을 만든다. 저장된 확률 값 들 중 가장 큰 확률 값을 가지는 행동을 상대 에이전트의 행동으로 예측하여 학습 에이전트의 행동 정책 수립을 위한 Q 학습에 포함한다. 자신의 Q 테이블은 상대 에이전트의 각 행동에 대한 학습 에이전트의 각 행동과의 연합 행동을 바탕으로 지금 상태에 대한 Q 값을 구한다.

본 논문에서는 Q 학습을 확장하여 적대적 멀티 에이전트 환경에 적합한 멀티 에이전트 Q 학습 방법을 제시한다. 특히 본 논문에서는 관찰되는 상대 에이전트의 행동을 나이브 베이지안을 통해 상대 에이전트의 행동선택함수인 상대방 정책 모델을 점진적으로 학습하고, 이 모델을 기초로 자신의 최적 정책을 학습하는 멀티 에이전트 Q 학습 방법을 제시한다. 그림 6은 나이브 베이지안을 이용한 상대 에이전트의 행동을 모델링 하여 자신의 행동 정책을 학습해 가는 과정이다.

행동 하나를 선택하는 것이 행동 주기(epoch)이며 고양이와 쥐의 위치가 같은 위치이면 하나의 에피소드(Episode)가 끝나게 된다. 그림 6에서 상대 에이전트(고양이)는 고정된 정책(쥐와 고양이의 위치가 가까운 쪽으로 이동)이나 동적인 정책(고양이의 행동 선택이 학습에 의한 경우)에 의해 선택된 행동을 실행하는 것을 학습 에이전트(쥐)는 관찰한다. 이 관찰된 상대 에이전트의 행동을 바탕으로 나이브 베이지안에 의한 상대 정책 모델을 생성하게 된다. 그리고 학습 에이전트는 행동 주기마다 생성된 상대 정책 모델에 의해 선택된 행동을 상대 에이전트가 수행할 것이라 예측한다. 이를 바탕으로 학습 에이전트는 멀티 에이전트 Q 학습을 수행하게 되며, Q 값에 의한 최적의 행동을 선택한다. Q 학습에 의해 학습 에이전트가 행동을 선택하게 되면 상대 에이전트도 행동을 동시에 수행하게 되고 보상 값(reward)을 받는다. 그리고 다음 상태로 바뀌게 되며 Q 테이블 값과 나이브 베이지안 모델을 업데이트한다.

상대 에이전트의 행동은 관찰할 수 있으며 하나의 행동 주기가 끝나면 행동에 대한 보상 값을 받게 된다.

학습 에이전트는 자신이 취할 행동 a_{self}을 식 1과 같이 계산하여 결정한다.

$$\pi[a_i|s] = \frac{e^{\overline{Q}(s, a_i)/\tau}}{\sum_{a_u \in A} e^{\overline{Q}(s, a_u)/\tau}} \qquad [식\ 1]$$

이때, τ는 행동 선택의 임의성(random)을 제어하는 변수이고, τ가 0에 가까울 경우에는 알려진 것 중 가장 Q값이 큰 행동이 선택되게 되고 τ가 클 경우에는 임의의 선택에 가깝게 행동이 선택된다.

$\overline{Q}(s, a_{self}) \equiv \sum_{a_{opponent} \in A_{opponent}} NBM(s, a_{opponent}) Q(s, a_{self}, a_{opponent})$는 상대방 정책 모델을 토대로 계산된 자신의 행동 a_{self}에 대한 Q-함수 기대 값이다.

에이전트는 식 2에 의해 선택된 자신의 행동 a_{self}을 실행한다. 이와 동시에 상대 에이전트도 행동 $a^*_{opponent}$을 선택하고 실행한다. 그 결과, 환경은 새로운 상태 s'로 변경되고 에이전트는 환경으로부터 보상 값 r을 받는다. 그러면 에이전트는 그림 4의 모델 갱신 식에 따라 상대방 정책 모델을

갱신하고, 또 식 2와 같이 Q 함수 값도 갱신한다.

$$Q(s, a_{self}, a^{*}_{opponent}) \leftarrow (1-\alpha)Q(s, a_{self}, a^{*}_{opponent}) + \alpha(r + \gamma \max_{a_{self}, a_{opponent}} Q(s', a'_{self}, a'_{opponent}))$$

[식 2]

여기서 $a'_{opponent} = \arg\max_{a'_{opponent}} NBM(s', a'_{opponent})$ 이다.

3장에서 제시한 고양이와 쥐게임에서 쥐의 보상 값 r은 식 3에 의해 주어진다.

$$r = r_1 + r_2 \qquad r_1 = \begin{cases} r_1 + 20 & (mouse.position = cheese.position) \\ 0 & (mouse.position \neq cheese.position) \end{cases}$$
$$r_2 = \begin{cases} r_2 + 1 & (mouse.position \neq cat.position) \\ 0 & (mouse.position = cat.position) \end{cases}$$

[식 3]

보상 값은 쥐가 고양이에게 잡히지 않고 치즈를 먹으면 많은 보상 값을 주도록 하였다. 그래서 쥐와 치즈와의 위치가 같으면 +20의 보상 값을 주고 쥐가 고양이에게 잡히지 않으면 계속 +1의 보상 값을 주도록 하였다. 20을 준 이유는 실험 결과 하나의 에피소드에서 쥐가 치즈를 먹기 위한 행동이 20번을 넘지 않기 때문이다.

```
MultiagentQLearning (state, joint_action)
/* state = {s0, s1, s2, ..., sn}, joint_action = {(a1, a1), (a1, a2),
..., (aself, aopponent)} */
/* aself : 학습 에이전트의 행동, aopponent : 상대 에이전트의
행동
/* 상태와 행동들 초기화 */
  Initialize for all s∈S, aself∈Aself and aopponent∈Aopponent;
  Q(sn, aself, aopponent) ← 0;
/* 에피소드가 끝날 때 까지 반복 수행한다. */
  Repeat for each state s∈S:
    /* 행동 선택 */
    Choose action aself and aopponent according to π(s, aself),
π(s, aopponent)
    /* 보상 값을 받고, 다음 상태로 전이한다. */
    Receive reward rself' , the action aopponent taken by the
opponent, and the next state s'
    /* 상대 정책 모델 갱신 */
    aNB ← nbValueCalculate(s, aopponent); //나이브
베이지안에 의한 확률 값 계산
    getBestAction(aNB); // 행동 확률 값이 가장 큰 행동
선택
    π(s, aopponent) ←getBestAction(aNB);

    /* Q 함수 갱신 */
  Q(s, a self, a*opponent) ← (1 − α)Q(s, a self, a*opponent)
    π(s, aself)← Q(s, a self, a*opponent)
```

(그림 6) 멀티 에이전트 Q 학습 알고리즘

6. 실험

6.1 실험 목적 및 방법

본 논문에서는 앞서 제안한 상대 정책 모델 기반의 멀티 에이전트 강화 학습의 효율과 에이전트의 성능을 분석하고 상대 정책 모델의 수렴을 분석하기 위해 고양이와 쥐게임을 이용한 실험을 전개하였다. 고양이와 쥐게임은 에피소드의 기간이 짧고 실행할 수 있는 행동이 제한적인 특징을 가지고 있다. 선행 연구인 관찰된 상대 행동을 기반으로 한 상대 모델 방법과 나이브 베이지안 확률 방법을 이용한 상대 모델 방법과의 비교 실험을 수행한다. 첫 번째 실험은 상대 에이전트가 고정된 정책을 가지는 경우에 대한 자신 에이전트의 단일 에이전트 강화 학습과 멀티 에이전트 강화 학습을 비교 실험한다. 이 실험을 통해 상대 에이전트를 고려하기 않는 단일 에이전트 강화 학습 방법보다 상대 에이전트를 고려하는 멀티 에이전트 강화 학습 방법이 더 효율적임을 보인다. 두 번째 실험은 나이브 베이지안 기반의 상대방 정책 모델이 강화 학습의 효율성에 미치는 영향을 분석한다. 상대 에이전트가 학습에 의해 정책을 변경하는 경우에는 상대 에이전트를 환경에서 분리하여 모델을 구축하는 것이 학습에 효과적일 것이다. 그러나 이 모델 수립 과정에서 학습 초기에 발생할 수 있는 자신의 최적 정책 수립에 영향을 주는 모델에 대한 정보가 제대로 이루어지지 않을 수 있다. 이를 방지하기 위해 나이브 베이지안 기반의 상대 모델을 수립함으로써 효율적인 강화 학습이 이루어진다고 볼 수 있다. 이를 위해 실험에서는 상대 에이전트(고양이)가 정

책을 변경하는 경우에 학습 에이전트(쥐)는 (i) 단순 Q 학습을 하는 경우와 (ii) 나이브 베이지안 기반 상대 정책 모델을 사용하여 Q 학습을 하는 경우에 대해 각각 10000 epoch를 수행한 후 비교 실험하였다. 그리고 각 경우의 실험에서 학습의 효율성을 분석하기 위해 식 4와 같은 벨만 오차(Bellman residual)를 측정해 보았다.

$$BE_t = \max_{s \in S} | V_t(s) - V_{t-1}(s) | \quad [식\ 4]$$

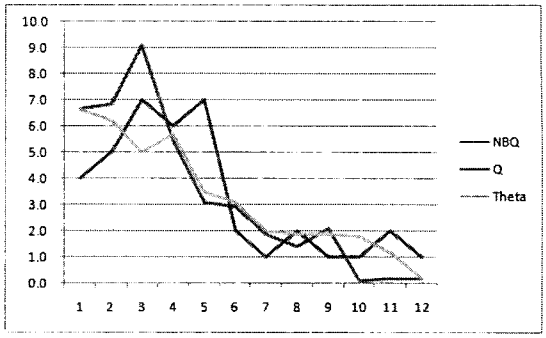
여기서 $V_t(s) = \max_{a_{self} \in A_{self}} \overline{Q}_t(s, a_{self})$이다.

세 번째 방법은 상대방 정책 모델이 학습의 결과로 나타나는 에이전트의 성능에 미치는 영향을 분석해 본다. 이 목적을 위해 선행 연구인 관찰된 행동 모델 기반의 Q 학습을 사용하는 경우와 본 논문에서 제안하는 나이브 베이지안 방법을 이용한 Q학습을 하는 학습 에이전트(쥐)가 고정된 정책을 수행하는 상대 에이전트와 게임을 차례대로 전개하면서 각 경우의 실험으로부터 게임 지속 시간(game duration time)을 측정하여 비교하였다. 게임 지속 시간은 새로운 게임이 시작되어 쥐가 고양이에 의해 잡힐 때까지 유지한 시간을 말한다. 게임 지속 시간이 길어지면 쥐의 성능이 향상되고 있는 것으로 판단할 수 있다.

세 번째 방법은 상대 에이전트가 정책을 변경하는 경우 학습 에이전트가 서로 다른 정책들을 사용할 경우에 따른 상대방 정책 모델의 정확성을 분석해 본다. 이를 위해 서로 다른 정책들(관찰된 행동 모델 기반의 Q 학습, 나이브 베이지안 기반의 Q 학습)을 사용하는 학습 에이전트와 고정된 정책을 사용하는 상대 에이전트 간의 게임들을 수행하면서 상대 에이전트가 실제 실행한 행동과 예측된 행동이 같은지를 정확성을 측정하는 기준으로 하였다.

6.2 실험 결과

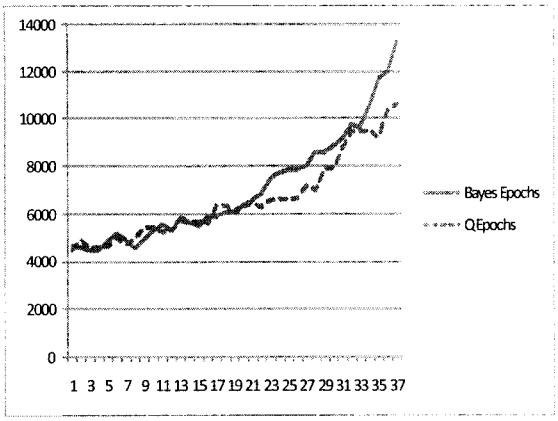
그림 7은 상대방 정책 모델이 학습 효율성에 미치는 영향을 분석하기 위한 비교 실험 결과를 나타



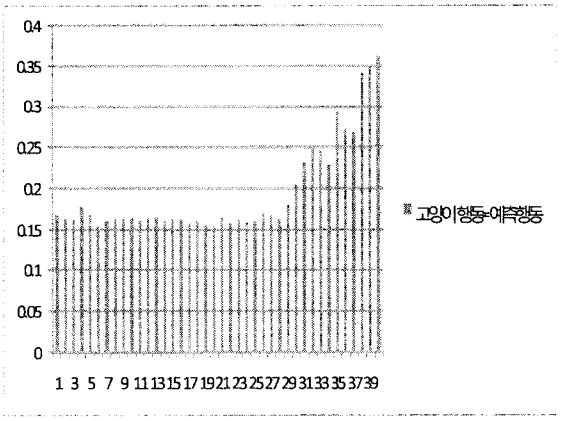
(그림 7) 벨만 오차 비교

내고 있다. 그래프의 가로축은 학습에 소요되는 시간을 반복주기(epoch)*1/1000로 나타내고 있고, 세로축은 학습시간에 따른 벨만 오차를 나타내고 있다. 그림 7을 통해 발견할 수 있는 중요한 사실들은 다음과 같다. 먼저, 전체적으로 그림 7의 경우, 즉 쥐가 상대인 고양이에 대한 행동 정책 모델을 이용하여 Q 학습을 하는 경우(Theta)보다 나이브 베이지안을 통한 상대 정책 모델을 가지고 학습하는 경우(NBQ)가 어떤 명시적인 상대 모델도 이용하지 않은 Q 학습(Q)보다 벨만 오차가 더 빨리 일정 수준 이하로 감소된다는 것이다.

벨만 오차의 감소는 곧 Q 함수의 수렴을 의미한다. 그림 7을 통해 발견할 수 있는 또 다른 사실은, 행동의 관찰 후 수립된 상대 정책 모델과 나이브 베이지안 기반 상대 정책 모델의 현저한 차이는 느낄 수 없지만 처음에 나이브 베이지안이 랜덤한 기준을 가지고 상대의 행동을 예측하기 때문에 벨만의 오차가 현저히 크지만 그 이후 급속한 속도로 오차의 범위가 줄어드는 것을 알 수 있다. 한편으로는 나이브 베이지안에 의한 확률이 너무 짧은 시간 안에 결정되어 거의 시작한 지 얼마 안 돼 Q 값이 수렴하는 것도 알 수 있었다. 예를 들어 나이브 베이지안을 이용하는 경우 5000 epoch 동안은 벨만의 오차가 현저히 큰 반면 14000 epoch 이후에는 다른 정책들보다 벨만 오차가 작은 것을 확인할 수 있다. 그리고 벨만 오차 5이하인 시점을 보면 고정된 정책을 사용하는 경우가 15000 epoch, 단순



(그림 8) 게임지속시간 비교



(그림 9) 모델 정확성

학습의 경우가 15000 epoch, 행동 정책 모델을 이용하는 경우가 14000 epoch이고 나이브 베이지안을 이용하는 경우가 13000 epoch로 나타나는 것을 볼 수 있다. 이를 통해 나이브 베이지안을 이용하는 경우가 상태에 대한 최적의 가치에 수렴해 간다는 것을 알 수 있다.

그림 8의 가로 축은 Epochs X 1000 으로 행동 주기(Epochs)는 하나의 행동 선택을 의미한다. 세로 축은 행동 주기의 발생 횟수를 나타낸다. 따라서 행동 주기가 증가한 다는 것은 게임의 지속적인 시간이 증가하고 있음을 의미한다. 초기에는 약간의 차이를 보이며 나이브 베이지안을 이용한 Q 학습의 행동 주기가 단순 Q 학습을 이용한 행동 주기보다 약간 길지만 별 차이는 없었다. 그러나 나이브 베이지안 방법을 이용한 Q 학습의 경우 22000 행동 주기에서 단순 Q 학습보다 행동 주기가 상당히 길어진 것을 확인할 수 있다. 이는 단순 Q학습을 이용하는 경우보다 나이브 베이지안을 이용한 Q학습의 수렴이 먼저 이루어지고 있음을 의미한다.

그림 9의 세로축은 고양이의 실제 행동과 예측 행동이 같을 확률을 나타내며, 가로 축은 epoch X (1/1000)을 나타낸다.

그림 9에서는 상대 에이전트인 고양이의 실제 행동과 나이브 베이지안 학습에 의한 행동의 일치에 대한 변화를 나타내고 있다. 그림 9의 실험 결과로부터 알 수 있는 사실은 하나의 행동이 실행할 수 있는 확률 즉 1/8인 0.125보다 나이브 베이지안 학습에 의한 하나의 행동이 실행할 수 있는 확률이 높다는 것을 알 수 있다. 이는 나이브 베이지안 학습 방법에 의한 행동이 실제 행동이 선택될 확률보다 높기 때문에 효율적인 모델 수립이 되었다고 볼 수 있다. 또한 그림 9의 결과에서 29000 epoch를 기준으로 확률 값이 상승하는 것을 볼 수 있는데, 이는 나이브 베이지안 학습에 의한 모델이 학습 에이전트가 상대 에이전트의 행동을 예측 할 수 있는 더 많은 기회를 가진다는 것을 의미한다. 그리고 정확성이 1이 되지 못하는 이유는 본 논문의 실험 환경이 비교적 짧은 에피소드이기 때문에 같은 상태를 만나지 쉽지 않기 때문이다.

7. 결론

본 논문에서는 적대적 멀티 에이전트 환경에서 상대 에이전트의 행동들에 의한 나이브 베이지안을 통해 상대 에이전트의 행동 정책 모델인 NBM을 학습하고, 이 모델을 바탕으로 다시 자신의 최적 정책을 학습하는 멀티 에이전트 강화 학습방법을 제시하였다. Q 학습 알고리즘을 확장한 이 멀티 에이전트 강화학습 방법은 상대 모델을 이용하는 기존의 멀티 에이전트 강화 학습 연구들에서 주로 시

도되었던 상대 에이전트의 Q 평가 함수 모델 대신 상대 에이전트의 행동 정책 모델을 비교적 간단한 형태의 나이브 베이지안을 통해 학습함으로써 학습의 효율성을 높인 것이 특징이다. 본 논문에서는 대표적인 적대적 멀티 에이전트 환경인 고양이와 쥐(Cat and Mouse) 게임을 테스트 베드로 삼아 다양한 비교 실험들을 전개하여 본 논문에서 제안한 나이브 베이지안에 기반을 둔 상대방 정책 모델 기반의 멀티 에이전트 강화 학습의 효과를 분석해보았다. 이 실험을 통해 나이브 베이지안을 통한 상대방 정책 모델을 이용하는 것이 강화 학습의 효율성과 에이전트의 성능 향상에 도움이 되며, 상대 에이전트가 고정 정책을 쓰는 경우는 물론 Q 학습을 하는 경우에도 나이브 베이지안을 통한 상대방 정책 모델 NBM의 수렴 성을 확보할 수 있다는 것을 확인하였다.

참 고 문 헌

[1] Yang E. and Gu D., "Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey," University of Essex Technical Report CSM-404, 2004

[2] Littman M.L., "Markov Games as Framework for Multi-Agent Reinforcement Learning," Proceedings of the 11th International Conference on Machine Learning, pp. 157-163, 1994

[3] Michael Bowling., "Multiagent Learning in the Presence of Agents with Limitations," Thesis of Doctor in University Carnegie Mellon, 2003

[4] J. Hu, M. P. Weliman, "Learning About Other Agents in a Dynamic Multiagent System," Journal of Cognitive Systems Research, Vol. 2, pp. 67-69, 2001

[5] M. L. Littman and C. Szepesvari, "A Generalized reinforcement learning models: Convergence and applications," In Proceedings of the 13th International Conference on Machine Learning, pp. 310-318, 1996

[6] Y. Gal and A. Pfeffer, "A Language for modeling agents' decision making processes in games," In AAMAS, 2003

[7] Birkendorf, A. Boker, H. U. Simon, "Learning Deterministic Finite Automata from Smallest Counterexamples," cCOLT TechReport, Universitat Dortmund, 1997

[8] A. Kautz and J. F. Allen, "Generalized plan recognition," In Proceedings of the National Conference on AI(AAAI), pp. 32-37, 1986

[9] Jan Ramon, Nico Jacobs, Hendrik Blockeel, "Opponent Modeling by analyzing play," Third International Conference on Computers Games (CG'02), Workshop on Agents in Computer Games, 2002

[10] D. Carmel, S. Markovitch, "Opponent Modeling in Multi-agent Systems," Lecture Notes in Artificial Intelligence, 1042, Springer-Verlag, 1996

[11] Aaron Davidson, Darse Billings, Jonathan Schaeffer, and Duane Szafron, "Improved opponent modeling in Poker," In proceedings of the International Conference on AI (ICAI-2000), pp. 1467-1473, 2000

[12] Bob Price, Craig Boutilier, "A Bayesian Approach to Imitation in Reinforcement Learning," IJCAI, pp. 712-720, 2003

[13] Kwun Han. and Veloso M., "Automated Robot Behavior Recognition Applied to Robotic Soccer," In Proceedings of IJCAI-99 Workshop on Team Behaviors and Plan Recognition, 1999

[14] Tesauro G., "Extending A-Learning to General Adaptive Multi-Agent Systems," In Advances in Neural Information Processing Systems 16, 2004

◐ 저 자 소 개 ◑

권 기 덕
1998년 경기대학교 전자계산학과(이학사)
2002년 경기대학교 대학원 전자계산학과(이학석사)
2003 ~현재 경기대학교 대학원 전자계산학과 박사과정
관심분야 : 강화학습, 멀티 에이전트 etc.
E-mail : kdkwon@kyonggi.ac.kr