2001년도 대한전기학회 하계학술대회 논문집 2001.7.18-20

연속적인 Q-학습을 이용한 자율이동로봇의 회피행동 구현

김민수
숭실대학교 전기공학과

Avoidance Behavior of Autonomous Mobile Robots using the Successive Q-learning

Min-Soo Kim
Dept. of Electrical Eng. Soongsil Univ.

Abstract - Q-학습은 최근에 연구되는 강화학습으로서 환경에 대한 정의가 필요 없어 자율이동로봇의 행동학습에 적합한 방법이다. 그러나 다개체 시스템의 학습처럼 환경이 복잡해짐에 따라 개체의 입출력 변수는 늘어나게 되고 Q함수의 계산량은 기하급수적으로 증가하게 된다. 따라서 이러한 문제를 해결하기 위해 다개체 시스템의 Q-학습에 적합한 연속적인 Q-학습 알고리즘을 제안하였다. 연속적인 Q-학습 알고리즘은 개체가 가질 수 있는 모든 상태-행동 쌍을 하나의 Q함수에 표현하는 방법으로서 계산량 및 복잡성을 줄임으로써 동적으로 변하는 환경에 능동적으로 대처하도록 하였다. 제안한 연속적인 Q-학습 알고리즘을 벽으로 막힌 공간에서 두 포식자와 한 먹이로 구성되는 먹이-포식자 문제에 적용하여 먹이개체의 효율적인 회피능력을 검증하였다.

1. 서 론

다수의 자율이동로봇 시스템(Multiple Autonomous Mobile Robot System)에서 각 개체는 주변환경의 인식 뿐만아니라 지속적인 환경변화에 적응할 수 있는 고도의 추론능력을 요구하고 있다[1][2][3][4]. 이렇듯 동적으로 변화하는 환경에 쉽게 적응하고 및 환경에 대한 응답으로써 개체의 행동이 적절하였는지에 대한 결정이 어느 정도 시간이 지난 경우에만 판단이 가능한 문제에서 개체들의 학습 방법으로서 비모델 강화학습인 Q-학습이 널리 사용되고 있다[5][6][7][8][9].

강화학습은 시간에 따른 개체의 행동에 대한 보상을 최대화하는 상태-행동 규칙 또는 행동전략을 찾는 것으로서 모델링하기 어려운 문제를 개념적인 용어들을 이용하여 쉽게 설계할 수 있는 장점이 있지만 상태와 행동을 정의하고 정책을 설계하기가 매우 힘들다는 한계를 갖는다. 또한 상태를 어떻게 설정하는지에 따라 학습시스템의 구조는 크게 달라지며 정책을 설정하는 방법에 따라 임무의 수행여부와 학습속도가 달라지게 되며 환경이 복잡해짐에 따라 상태의 수가 증가하게 되어 계산량 및 복잡도는 기하급수적으로 증가하게 된다.

개체가 주어진 상황에 따른 행동을 학습하기 위해서는 환경 및 개체들에 대한 정보를 상태변수로 사용해야하는데, 상태변수가 하나 증가할 때마다 상태-행동 쌍의 수는 기하급수적으로 증가는 문제를 낳게 된다. 따라서 이러한 문제를 해결하기 위해 Q함수를 연속적으로 사용하여 계산량을 줄이는 방법인 연속적인 Q-학습(SQLA; Successive Q-Learning Algorithm)을 제안하였으며, 이를 먹이-포식자 문제(Prey and Predator Problem)에 적용하였다[2][10]. 먹이-포식자 문제는 두 포식자가 한 먹이를 추적하는 문제로서 먹이개체는 두 포식자 및 장벽을 피하도록 학습해야 한다.

2절에서는 Q-학습 및 제안한 SQLA에 대해 살펴보고 3절 시뮬레이션 및 결과에서는 SQLA를 두 포식자와 한 먹이로 구성된 먹이-포식자 문제에 적용하여 그 성능을 검증하였고 4절에서는 결론을 맺었다.

2. Q-학습 및 SQLA

2.1 Q-학습

Q-학습은 강화학습의 일종으로써 환경에 대한 모델링 없이 실시간 학습이 가능하기 때문에 동적인 환경변화에 따라 개체행동을 결정하는데 적합한 학습방법이다. 그림 1은 Q-학습의 기본 구조를 나타낸다[6][9]. 주위환경에 대한 반응으로서 개체(Agent)는 어떤 행동을 취하게 되고 그 행동은 환경에 영향을 주게 된다. 또한, 그 행동에 대한 결과로서 개체는 환경으로부터 보상(Reward)을 받게 되는데, 이 보상값은 개체의 행동이 현재의 학습 흐름에서 어느 정도 좋게 작용했는지에 따라 다르게 주어지는 강화신호이며, 이 정보에 기초하여 학습시스템은 개체의 행동 규칙을 갱신시키게 된다.

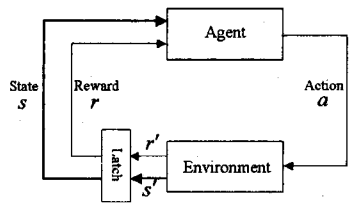


그림 1. Q-학습의 기본구조

Q-학습의 목적은 각 상태에서 감쇄된 미래의 보상값에 대한 기대값을 최대화하는 상태-행동규칙을 찾는 것이다. 즉, 상태 s에 대한 최적 행동 a을 찾는 것이 목적이 된다. 강화학습 알고리즘은 실제 가치함수인 $Q(s,a)$의 추정값에 기초하는데, 이 추정값은 상태 s에서 행동 a를 수행하였을 경우에 주어지는 미래 강화신호의 감쇄된 합에 대한 기대값으로 주어진다. 결정적인 보상과 행동인 경우, 다음 상태 s'와 다음 상태에서의 행동 a'가 주어졌을 때 Q-학습을 통해 Q값은 식 (1)처럼 갱신된다.

$$Q(s,a) \Leftarrow r + \gamma \max_{a'} Q(s',a') \qquad (1)$$

비결정적인 보상과 행동인 경우, Q값의 갱신은 식 (2) 처럼 주어진다.

$$Q(s,a) \Leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \qquad (2)$$

단, r은 강화신호값이며 α와 γ는 각각 학습상수 및 감쇄인자로서 α는 현재의 Q값에 대한 학습속도를 결정하고 γ는 미래 상태로부터의 현재 Q값에 미치는 영향정도를 나타낸다.

Q-학습을 단계별로 구분하면 8단계로 나눌 수 있는데 그 순서는 아래와 같다.

단계1. 임의의 값으로 $Q(s,a)$값을 초기화
단계2. 실행환경의 초기화 및 초기상태 s 결정
단계3. Q함수 정책에 따라 상태 s에 따른 행동 a 선택.
단계4. 단계3의 결과로서 보상 r, 다음상태 s', 그리고 다음행동 a' 결정
단계5. 식(2)에 기초하여 $Q(s,a)$값 갱신
단계6. 다음상태 s'를 현재상태 s로 변경
단계7. 종료상태에 도달할 때까지 단계3부터 반복
단계8. 미리 설정한 회수만큼 단계2부터 단계7 반복

Q-학습에서의 특징은 시행착오(Trial and Error)에 따른 학습과 지연보상(Delayed Reinforcement)라 할 수 있다. 그러나 Q-학습은 상태-행동규칙에 기초하여 학습하기 때문에 상태가 증가하게 되면, $Q(s,a)$쌍이 기하급수적인 증가한다는 문제가 존재하게 된다. 특히, 이러한 경우는 다개체시스템과 같이 환경이 복잡해질 경우에 발생하게 된다. 따라서 이러한 문제를 해결하기 위한 방법으로 SQLA 방법을 제안하였다.

2.2 SQLA

SQLA는 다개체 시스템과 같이 많은 개체들이 서로 상호작용을 통해 행동을 결정해야 하는 경우, 개체는 다른 개체의 행동 및 환경에 능동적으로 대처하기 위해서 다변수 입력을 필요로 하게 된다. 그러나, 일반적인 Q-학습 방법은 상태에 따른 행동 쌍으로 주어지는 Q값을 갱신시키는 방법으로 개체의 행동을 학습시키기 때문에 상태변수의 증가는 상태-행동 쌍의 기하급수적인 증가를 낳게 되어 계산량 및 학습시간이 기하급수적으로 증가하게 된다. 이러한 변수의 증가에 따른 계산량의 증가를 줄이기 위해 SQLA를 제안하였다.

SQLA는 Q함수를 여러개로 분할하는 방법이며 분할된 Q함수의 출력을 다음 단에 위치하는 Q함수의 입력으로 사용함으로써 계산량을 줄이는 방법이다.

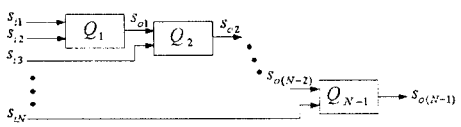


그림 2. SQLA의 입출력 구조

그림 2는 Q함수가 2입력 1출력을 갖는 경우, N개의 입력에 대한 SQLA의 출력을 나타내었다. 변수 s_{ij}(단, $j=1,2,\cdots,N$)는 j번째 입력변수로서 앞첨자 i는 입력변수임을 의미한다. 그리고 변수 s_{ok}(단, $k=1,2,\cdots,N-2$)는 k는 k번째 변수를, o는 출력을 나타낸다. 마지막으로 Q_m는 분할된 m번째 Q함수를 의미한다.

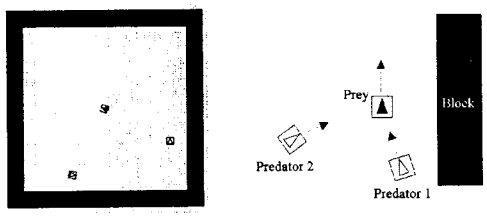
예를 들어, 두 포식자가 한 먹이를 추적하는 먹이-포식자 문제 중에서 먹이의 행동에 대한 학습에 대한 살펴보면, 먹이가 포식자들 및 장벽에 대한 회피행동을 위해서는 포식자1에 대한 정보와 포식자2에 대한 정보, 그리고 장벽에 대한 정보를 각각 입력변수로 사용해야 되는데, 이 경우 입력변수의 수인 N은 3이 된다. 따라서 Q함수는 포식자1,2에 대한 정보를 입력으로 사용하는 Q_1과 Q_1의 출력 및 장벽에 대한 정보를 입력으로 사용하는 Q_2로 나누어지게 된다.

Q함수의 학습은 비결정적인 보상과 행동인 경우에 사용하는 식 (2)에 기초하여 학습하였는데, Q_1의 출력 중에서 상태에 따른 행동값 중에서 최대인 Q_1값을 Q_2의 입력으로 사용하였다. 이 경우 Q_1은 Q_2에 비해 선택의 경우가 많게 된다.

3. 먹이-포식자 문제

먹이-포식자문제는 닫힌 공간에서 한 먹이와 두 포식자가 서로 회피와 추적을 목적으로 서로 상반된 행동을 취하는 문제를 말한다. 이 논문에서는 장벽과의 충돌 없이 두 포식자의 추적을 가장 잘 회피하기 위한 먹이의 회피행동을 학습시키는 문제로 설정하였다.

포식자 개체는 다른 포식자의 행동을 고려하지 않고 행동하는데, 먹이를 발견하게 되면 최단거리로 먹이를 향해 추적한다. 먹이 개체는 두 포식자의 추적 및 장벽을 회피하도록 행동하는 것이 목적이다. 따라서 먹이개체는 포식자들에 대한 상대적인 방향과 거리정보 그리고 장벽에 대한 방향 및 거리 정보를 이용하여 행동하여야 한다. 즉, 먹이개체의 학습을 위해서는 두 포식자에 대한 정보와 장벽에 대한 정보를 최적의 회피행동을 결정하도록 입력으로 사용하여야 한다. 먹이는 입력변수로 포식자들의 위치 및 장벽의 위치에 대한 정보를 사용하고 출력변수로는 먹이의 회피 행동이 사용되는데, 포식자 및 장벽의 위치를 파악하기 위해서는 상대적인 방향정보와 거리정보가 필요하며 먹이가 회피행동을 하도록 이동하기 위해서는 이동해야 할 방향과 속도(또는 거리)가 필요하게 된다. 이때 방향에 대한 정보는 모두 먹이 개체가 향하던 방향이 기준이 되어 상대적인 회전방향(전,후,좌,우)을 사용한다.



(a)시뮬레이션 환경 (b) 먹이의 회피행동 예

그림 3. 먹이 개체의 회피행동

그림 3에서는 먹이개체가 포식자들의 추적 및 장벽을 피해서 이동하는 경우를 보여주고 있다. 먹이의 행동은 전 상태의 방향을 기준으로 포식자와 장벽의 방향 및 거리를 계산한 후 최적의 위치로 이동하게 된다.

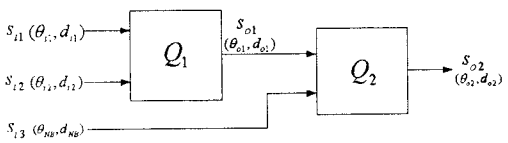


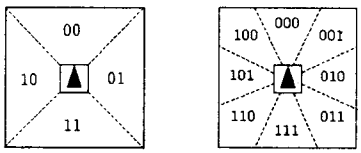
그림 4. 먹이-포식자 문제에서 Q함수의 입출력 변수

그림 4에서는 먹이개체가 포식자 및 장벽을 회피하도록 학습하기 위해 사용된 SQLA의 입출력 변수를 나타내었다. Q_1및 Q_2의 입출력 상태변수인 θ 및 d에 주어지는 아래첨자 i는 입력을, 그리고 o는 출력을 의미한다. 학습결과의 평가함수인 Q함수에는 학습 중에 받은 보상값들이 누적되어 있어 각 상태에서 많은 보상을 받은 행동일수록 좋은 행동으로 분류되어 선택되게 된다. 그리고 θ_{NB}와 d_{NB}는 먹이 개체의 진행방향으로부터 가장 가까운 장벽(the Nearest Block)의 방향과 거리를 각각 나타낸다.

4. 시뮬레이션 및 결과

포식자는 벽과 먹이의 방향 및 거리를 감지한 후 먹이를 추적하여 포획하는 것이 목적이며, 먹이는 가장 가까운 벽과 두 포식자의 방향 및 거리를 감지한 후 최적의 위치로 회피하는 것이 목적인 문제이다.

그림 5-(a)는 4방향을 갖는 상태변수들인 θ_{i1}, θ_{i2}, θ_{NB} 그리고 θ_{o1}에 대한 상태값으로서 2비트로 코딩하였으며, (b)는 먹이가 이동해야할 방향을 결정하는 Q_2의 출력인 θ_{o2}의 설정 가능한 값을 나타내고 있다. 최종 출력값인 θ_{o2}는 다른 θ보다 정밀한 값을 필요로 하기 때문에 8방향인 3비트로 코딩하였다.



(a) 4방향인 경우 (b) 8방향인 경우

그림 5. 상태변수 θ의 정의

거리인 d_{i1}, d_{i2} 그리고 d_{NB}에 대한 상태값은 1미터 미만일 경우 '00', 2미터 미만일 경우 '01', 그리고 3.5미터 이상일 경우 '10'이며, 그 이외에는 '11'로 구분되는 4가지 경우에 대해 2비트로 코딩하였고, Q_2의 출력인 d_{o2}경우에는 '00'이면, 이동하지 않고, '01'이면 최대이동거리의 33%, '10'이면 최대이동거리의 66%, 그리고 '11'이면 최대이동거리만큼 이동하도록 설정하였다.

마지막으로 Q_1의 출력인 θ_{o1}과 d_{o1}은 Q_2의 입력과 동일하게 사용되기 때문에 은닉상태(Hidden States)이며 다른 입력 변수와 동일하게 2비트로 코딩하였다.

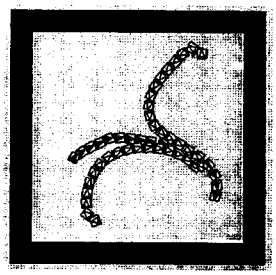


그림 6. 포식자에 대한 회피능력

그림 6은 포식자들의 추적에 대한 먹이의 회피 행동을 시간에 따라 표현한 것으로 포식자1은 우측 상단에서, 포식자2는 좌측 하단에서 먹이를 추적하기 시작한다. 먹이의 회피능력을 검증하기 위해서 포식자 개체 및 먹이 개체의 최대 이동거리를 0.5미터로, 그리고 최소 이동거리를 0.05미터로 각각 제한하였다. 또한 개체의 회전 반경을 60도로 제한을 두어 먹이 및 포식자 개체 모두에 적용하였다. 시뮬레이션 결과 두 포식자는 최단거리로 먹이를 추적하고 먹이는 포식자의 위치 및 장벽을 고려하여 회피행동을 보임을 알 수 있다.

4. 결 론

동적인 환경에서 개체의 행동을 다루는 문제에 있어서 Q-학습이 좋은 성능을 발휘함은 이미 알려져 있다. Q-학습은 상태-행동 쌍으로 정의되는 Q함수에 의해 개체의 행동을 정의하고 개체의 행동결과에 보상을 줌으로써 Q함수를 갱신시키는 학습알고리즘이다. 그러나 한 개체가 아닌 다개체시스템 처럼 복잡한 환경인 경우에는 각 개체들이 서로 상호작용을 통해 개체의 행동을 결정해야 하기 때문에 시스템이 복잡해지고 계산량이 증가하게 된다. 따라서 이러한 다개체시스템과 같이 복잡한 주위 환경에 적합한 SQLA를 제안하였으며, 이를 먹이-포식자 문제에 적용하여 먹이개체가 장벽 및 포식자의 추적에 효율적으로 회피함을 검증하였다.

[참 고 문 헌]

[1]김민수 외 2인 , "*클러스터링에 의한 자율이동 로봇의 군지능 알고리즘 구현*," 전기학회 하계학술대회 논문집, Vol. G, pp. 2293-2295, 1997

[2]김민수 외 3인, "*협조행동을 위한 자율이동로봇의 강화학습에서의 먹이와 포식자문제*", 2000년도 대한 전기학회 추계학술대회 논문집 (서울대학교), Vol. D, pp. 809-811, 2000

[3]Craig Boutilier, Thomas Dean and Steve Hanks, "Decision Theoretic Planning: Structural Assumptions and Computational Leverage," *JAIR (Journal of AI Research), 1999*

[4]Z. Ghahramani and M. Jordan, "Factorial Hidden Markov Models," *Machine Learning* Vol. 29, pp. 245-273, 1997

[5]C.Watkins, P. Dayan, Q-Learning, *Machine Learning* Vol. 8, pp. 279-292, 1992

[6]Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore,"Reinforcement Learning: A Survey". *JAIR (Journal of AI Research)*, Vol. 4, 1996

[7]Leslie Pack Kaelbling, Michael L. Littman and Anthony R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains", *Artificial Intelligence*, Vol. 101, 1998

[8]L. Rabiner "A tutorial on Hidden Markov Models and selected applications in speech recognition", *Proc. IEEE* 77(2) pp:257-286, 1989

[9]Richard Sutton and Andrew Barto, *"Reinforcement Learning: An Introduction"*, MIT Press, 1998

[10]Andrea Bonarini, "Reinforcement Learning of Hierachical Fuzzy Behaviors for Autonomous Agents," Proceedings of IPMU96, pp. 1223-1228, 1996

[11]Y. Bengio, Markovian Models for Sequential Data, *Neural Computing Surveys* 2, pp. 129-162, 1999

[12]Dimitri P. Bertsekas and John Tsitsiklis, *"Neuro-dynamic programming"*, Athena Scientific, 1996

[13]F. Jelinek, *"Statistical Methods for Speech Recognition"*, MIT Press, 1997