Learning Classifier System을 이용한 행동 선택 네트워크의 학습

윤은경[O], 조성배
연세대학교 컴퓨터과학과
ekfree@candy.yonsei.ac.kr[O], sbcho@cs.yonsei.ac.kr

Learning Action Selection Network Using Learning Classifier System

Eun-Kyung Yun[O], Sung-Bae Cho
Dept. of Computer Science, Yonsei University

요 약

행동 기반 인공지능은 기본 행동들의 집합으로부터 적절한 행동을 선택함으로써 복잡한 행동을 하도록 하는 방식이다. 행동 기반 시스템은 1980년대에 시작되어 이제는 많은 에이전트 시스템에 사용되고 있다. 본 논문에서는 기존의 P. Maes가 제안한 행동 선택 네트워크에 Learning Classifier System을 이용한 학습 기능을 부가하여, 변하는 환경에 적절히 적응하여 행동의 시퀀스를 생성할 수 있는 방법을 제안하다. 행동 선택 네트워크는 주어진 문제에 따라 노드 간 연결을 설계자가 미리 설정하도록 하는데, 해결해야 할 문제가 변함에 따라 네트워크에서의 연결 형태가 변형될 필요가 있다. Khepera 로봇을 이용한 시뮬레이션 결과, 행동 선택 네트워크에서의 학습이 유용함을 확인할 수 있었다.

1. 서 론

로봇 에이전트에서의 학습은 센서의 불확실성, 주위 환경에 대한 부분적인 관찰, 유동적인 환경 등의 원인으로 인해 매우 어려운 문제로 알려져 있다[1]. 행동 기반 시스템[2]에서의 학습은 특히 불확실성과 유동성을 동시에 갖고 있는 환경에 직면하고 있기 때문에 오랜 시간에 걸친 최적화보다는 짧은 시간 내 개선된 효율성에 초점을 맞추고 있다. 효율적인 행동 선택은 행동 기반 시스템에 있어서 주 관심사이다. 이 문제는 널리 사용되고 있는 강화 학습으로 쉽게 구조화될 수 있다. 이와 관련된 예로는 1990년 P. Maes와 R. Brooks[3]의 hexapod walking, 1991년 S. Mahadevan과 J. Connell[4]의 box-pushing 등이 있다. 또한 멀티 로봇 시스템에서는 M. Dorigo와 M. Colombetti[5]가 기존의 강화 학습을 수정하여 Shaping 의 개념을 도입하기도 하였다.

본 논문에서는 P. Maes가 제안한 행동 선택 네트워크에 Learning Classifier System(LCS)을 이용하여 자연계와 유사하게 환경 변화에 적응하도록 하는 모델을 제안한다. 기존의 행동 선택 네트워크가 갖고 있는 구조 수정의 어려움을 해결하기 위해 학습 기능을 추가하여 동적인 환경에 적합한 네트워크가 구축되도록 한다.

2. 행동 선택 네트워크

P. Maes가 제안한 행동 선택 네트워크[6]는 기본 행동 노드와 외부 환경 상태, 내부 목표가 서로 연결되어 상호 협력과 억제를 수행한다. 노드는 하나의 기본 행동을 나타내며 선행 조건, 추가 조건, 삭제 조건, 활성도, 실행 코드로 구성된다. 그림 1은 하나의 노드를 구성하는 요소들을 나타내고 있다.

외부 상태가 행동의 선행 조건일 때 외부 상태 연결이 설정된다. 목표에 도움이 되는 행동일 경우, 목표와 행동 사이에 목표 연결이 설정된다. 목표에 방해가 되는 행동일 경우

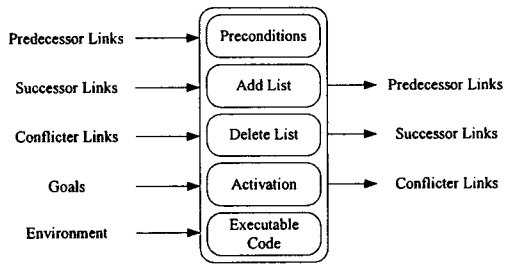


그림 1. 행동 선택 네트워크의 노드 구성 요소

억제 목표 연결이 설정된다. 행동들 사이에는 선행자 연결, 후임자 연결, 억제자 연결이 존재한다. 그림 2는 각 연결에 대한 조건을 보여준다.

행동 선택의 과정은 다음과 같은 순서로 이루어진다.

1. 외부 상태와 목표로부터 들어온 신호를 행동에 입력한다.
2. 행동들 사이의 내부연결을 통해 활성도를 교환한다. 행동들의 활성도가 무한히 커지는 것을 막기 위해 정규화를 수행한다.
3. 행동들 중에서 선행 조건이 모두 참이고, 활성도가 임계값보다 클 경우 선택한다.

3. LCS에 의한 행동 선택 네트워크의 학습

3.1 LCS (Learning Classifier System)

LCS는 1978년 Holland와 Reitman[7]에 의해 제안된 시스템으로, classifier라 불리는 규칙들을 크레딧 할당 시스템과 규칙 발견 시스템을 통해 학습시킨다. 대략적인 구조는 그림 3과 같다. Classifier system은 여러 개의 규칙들,

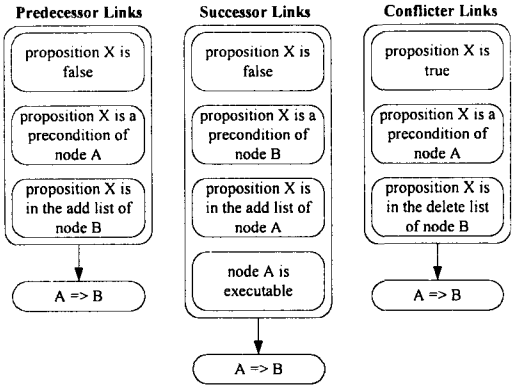


그림 2. 내부 연결 조건

즉 classifier로 구성된다. 하나의 classifier는 조건부와 실행부로 나눌 수 있는데, 조건부는 {0,1,#}, 실행부는 {0,1}로 구성된다. #은 0 또는 1과 모두 매칭된다. LCS는 그림 3과 같이, 크게 세 부분으로 구성된다.

- Classifier System: 외부로부터 입력된 메시지와 모든 classifier의 조건부를 비교하여 매칭되는 것을 찾는다. 매칭된 classifier들은 경쟁을 통해 하나의 승자만이 선택되어 외부로 보내진다. 경쟁에는 bid라는 값이 사용된다.

$$bid = c * specificity * strength$$

(*c*: 1보다 작은 양수, *specificity*: 조건부에서 #이 아닌 비트 수, *strength*: 이 값이 클수록 경쟁에서 이길 확률이 높음)

외부로 보내진 액션에 대해 환경으로부터 피드백이 들어 오면 승자 classifier의 strength가 조정된다.

- Credit Assignment System: 이 과정에서의 주 역할은 규칙의 유용성에 따라 규칙들을 분류하는 것이다. 외부로부터 들어온 메시지가 어떤 출력을 내보내면 그에 맞는 보상이 내부로 전해진다. 실제 출력을 내보내는 데에 관련이 있는 다른 classifier들도 그 보상을 나누어 받게 된다. 전통적으로 bucket brigade 알고리즘을 사용한다.
- Rule Discovery System: 규칙 발견을 위해 유전자 알고리즘이 사용된다. 유전자 알고리즘은 최적화나 규칙 발견의 목적으로 많이 사용되는 확률적 알고리즘이다. 이는 주어진 문제에 대한 해집합을 수정하는 작업을 한다. LCS에서는 classifier가 해의 역할을 한다.

3.2 LCS를 이용한 학습

LCS는 유동적인 환경에 적합한 규칙들을 학습을 통해 생성하는 기계 학습 방법으로, 외부 환경의 변화나 에이전트에게 주어진 과제의 변화에 따라 적절한 행동 선택 네트워크 구조를 생성하는 데 적합한 방법이라 할 수 있겠다. 행동 선택 네트워크는 초기 연결이 고정되어 있기 때문에 변화에 둔감하다는 단점이 있다. 각 노드 간 연결을 학습을 통해 주어진 과제나 환경에 따라 적절한 연결이 설정된다면 보다 효율적인 행동 선택을 할 수 있을 것이다.

본 논문에서는 행동 네트워크 구조의 학습을 위해 LCS를 적용하여 로봇의 과제 해결력 향상을 확인해보고자 한다. 먼저 학습을 위해 LCS의 기본 요소인 규칙을 어떻게 정의할 것인지 결정해야 한다. 규칙은 상태 노드와 과제 정도, 각 노드 간 연결들로 구성된다(그림 4). 기존의 행동 네트워크에서는 행동 노드 간 연결이 세 종류였으나, 본 논문에서는 유사한 기능을 갖는 선행자 연결과 후임자 연결을 흥분성 연결로 정의하고 억제자 연결을 그대로 억제성 연결로 정의한다. 또한 LCS에서

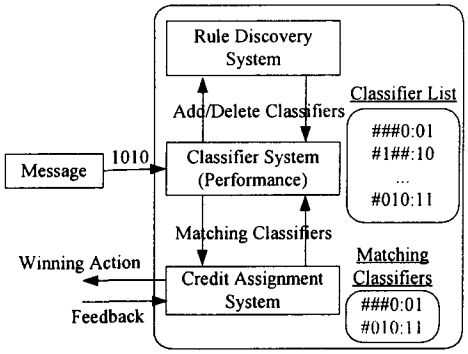


그림 3. LCS의 구조

의 초기 classifier 집합은 설계자가 과제 목적에 따라 정하기로 한다.

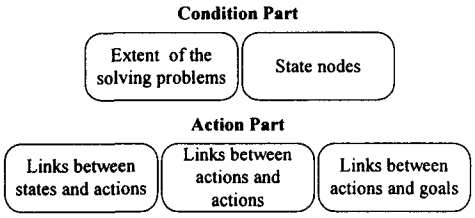


그림 4. 규칙의 구성

4. 실험 및 결과

4.1 실험 설계

에이전트가 이미 구축되어 있는 행동 선택 네트워크를 가지고 과제를 수행하되, 과제가 변함에 따라 네트워크 구성을 수정하도록 노드 간 링크를 학습시키고자 한다. 에이전트에게 주어진 과제는 특정 목적지까지 최단 경로로 도달하는 것이다. 본 논문에서는 행동 네트워크를 초기에 장애물 피하기와 직진하기 행동만 하도록 설정한다. 따라서 도달하고자 하는 목적지의 위치와 관계없이 로봇은 직진하기와 장애물 피하기 행동만을 이용하여 과제를 수행하려 할 것이다. 하지만 그림 5에서처럼 로봇의 위치에 따라, 직진하기보다는 새로운 행동이 더 유리하다. 초기 네트워크를 규칙으로 변환하여 classifier list를 구성하고 LCS를 이용한 학습을 통해 에이전트의 위치가 변하더라도 최단 경로로 갈 수 있는 행동을 선택하도록 하고자 한다. 초기 행동 네트워크는 그림 6과 같이 구성하고, 실험을 위해 Khepara 로봇 시뮬레이터[8]를 사용한다.

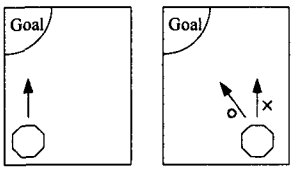


그림 5. 에이전트의 위치에 따른 행동 선택

4.2 실험 결과

본 연구에서는 에이전트가 학습을 통해 최단 경로로 목적지에 도달하는 것을 목표로 삼았다. 로봇의 목적지는 왼쪽 상단 모서리이며, 학습 과정 및 테스트 시 로봇의 초기 위치는 임의로 정하였다. 그림 7에서 볼 수 있듯이 로봇은 초기 행동 네트워크

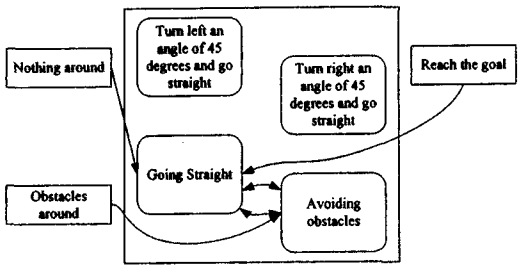


그림 6. 초기 행동 선택 네트워크의 구성(내부 연결의 경우, 점선은 억제성 연결, 실선은 흥분성 연결)

구성에 따라 처음에는 직진하기 행동을 많이 보였지만 학습을 거친 후 움직임에서는 직진하기 이외의 행동이 초반에 나타남을 볼 수 있다. 그림 8은 학습 전후 행동 네트워크의 한 예이다. 초기 행동 선택 네트워크(그림 6)와 비교해 보았을 때 학습 과정을 거친 후, 각 노드 간 연결이 바뀌었음을 확인할 수 있다. 그림 9는 각 epoch에서 에이전트가 올바른 행동을 선택한 평균 횟수를 나타낸 것이다. 시간이 지남에 따라 학습을 통해 적절한 행동을 선택했음을 보여주고 있다. 표 1은 에이전트가 목적지에 도달한 평균 스텝 수를 나타낸 것이다. 학습 후에 상당히 빠른 시간 안에 목적지에 도달함을 확인할 수 있다.

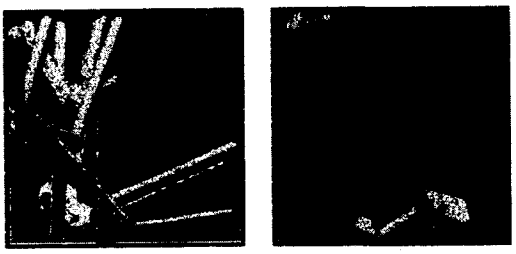
표 1. 에이전트가 목적지에 도달하는 평균 스텝 수

	Before learning	After learning
Average steps	7936.7	4421.8

5. 결론

본 논문에서는 P. Maes가 제안한 행동 선택 네트워크[10]에 학습의 개념을 도입하였다. 에이전트가 변하는 상황에 적절한 행동을 선택하기 위해서는 변경된 조건들을 고려하여 그에 맞는 행동을 택해야 할 것이다. 따라서 유동적인 환경에 적응적인 규칙들을 생성해내는 규칙 기반 시스템의 학습 방법인 LCS를 행동 선택 네트워크 학습 방법으로 선택하여 적용해 보았다. 그 결과 에이전트의 변하는 상황에 알맞은 규칙들이 생성됨을 확인할 수 있었으며 에이전트의 움직임 역시 상당히 만족할 만한 수준이었다.

하지만 본 논문에서는 행동 네트워크 학습의 유용성을 보이기 위해 행동이 네 가지로 구성되어 있는 매우 간단한 모델을 사용하였다. 따라서 이보다는 좀 더 다양한 행동 노드를 갖춘, 복잡한 행동 네트워크를 구성하여 변화에 적응 가능한 행동 네트워크 구축에 대한 연구가 진행되어야 할 것이다.



(a) Before learning (b) After learning

그림 7. 에이전트의 초기 위치에 따른 행동 네트워크 학습 결과 (파란 원은 에이전트의 초기 위치를 나타냄)

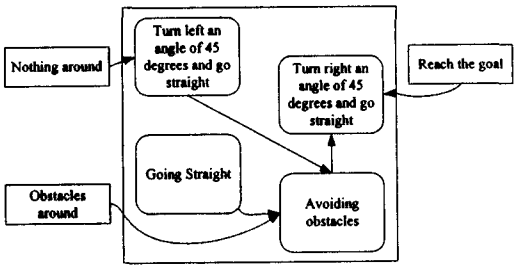


그림 8. 학습 후 행동 선택 네트워크 구성의 변화 (내부 연결의 경우, 점선은 억제성 연결이고 실선은 흥분성 연결)

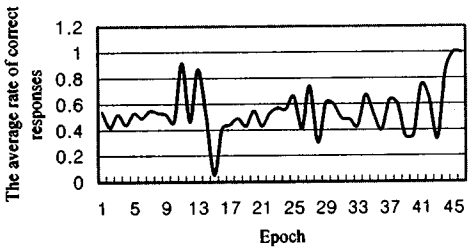


그림 9. 각 epoch에서 옳은 행동의 평균 수행 횟수

감사의 글

이 논문은 한국학술진흥재단의 연구과제(2002-005-H20002)에 의해 지원되었음.

참고문헌

[1] M. J. Mataric, "Learning in Behavior-Based Multi-Robot Systems: Policies, Models, and Other Agents," *Journal of Cognitive Research*, vol. 2, no. 1, pp. 81-93, 2001.

[2] R. Arkin, *Behavior-Based Robotics*, MIT Press, Boston, MA, 1998.

[3] P. Maes, and R. Brooks, "Learning to Coordinate Behaviors," *Proceedings of AAAI-90*, pp. 796-802, Boston, MA, 1990.

[4] S. Mahadevan, and J. Connell, "Scaling Reinforcement Learning to Robotics by Exploiting the Subsumption Architecture," *Eighth International Workshop on Machine Learning*, Morgan Kaufmann, pp. 328-337, 1991.

[5] M. Dorigo, and M. Colombetti, *Robot Shaping: An Experiment in Behavior Engineering*, MIT Press, Cambridge, MA, 1997.

[6] T. Tyrrell, " An Evaluation of Maes' Bottom-up Mechanism for Behavior Selection," *Adaptive Behavior*, vol. 2, pp. 307-348, 1994.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, MA, 1989.

[8] O. Michel, "Khepera Simulator Package version 2.0: Freeware Mobile Robot Simulator," *http://diwww.epfl.ch/lami/team/michel/khep-sim*, 1996.