Reinforcement Learning을 이용한 턴이 바뀌지 않는 경우가 있는 보드게임의 최적해 탐색

정현수[O1] 장하영[2] 장병탁[2]

경기과학고등학교[1] 서울대학교 컴퓨터공학부[2]

hsjeong1121@gmail.com, hyjang@bi.snu.ac.kr, btzhang@bi.snu.ac.kr

Optimal strategy for A Board Game with Nonconsecutive Turns Using Reinforcement Learning

Hyeon-Su Jeong[O1] Ha-Young Jang[2] Byoung-Tak Zhang[2]

Gyeonggi Science High School for the Gifted[1]

Dept. of Computer Science and Engineering, Seoul National University[2]

요 약

사용자가 교대로 게임을 진행하는 턴제 보드게임은 규칙이 단순하거나 말판의 크기가 그리 크지 않은 경우에는 트리 형태의 탐색 알고리즘을 통하여 게임의 전략을 개발하는 경우가 일반적이다. 하지만 특정 상황에 따라서 사용자의 턴이 바뀌지 않는 경우가 있는 보드 게임의 경우에는 일반적인 경우보다 문제 공간의 크기가 급격히 증가하기 때문에 이러한 방식을 사용하는데 어려움이 있게 된다. 따라서 본 논문에서는 이러한 경우에 강화 학습 기법을 이용하여 방대한 문제공간에서의 최적해를 보다 효율적으로 찾아내는 기법에 대하여 연구를 진행하였다. 이를 위하여 제안한 방법론을 이용하여 설계한 전략과 기존의 트리기반 탐색 알고리즘을 이용하여 구현한 전략과의 성능 비교를 통해서 제안한 방법론이 보다 큰 문제 공간에서도 효율적으로 최적해를 탐색할 수 있음을 확인하였다.

1. 서 론

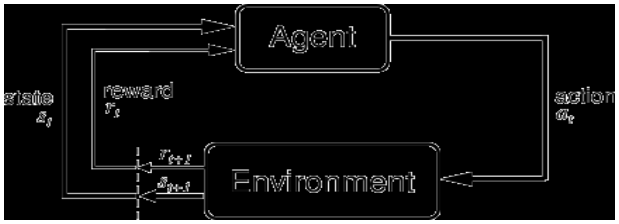
보드 게임은 체스, 바둑, 오셀로와 같이 일정하게 턴이 바뀌는 보드게임과 그렇지 않은 보드 게임으로 나눌 수 있다. 턴이 일정하게 유지되는 턴제 보드게임(turn-based strategy game, TBS)의 인공지능(Artificial Intelligence, 이하 AI) 제작에는 트리의 깊이에 따라 턴을 구해내기 용이한 알파베타 가지치기(Alpha-Beta Pruning), 미니맥스 알고리즘(Minimax Algorithm)과 같은 트리 형태의 알고리즘이 쓰인다. 이와 반대로 트리의 깊이에 따라 턴을 구해내기 어려운 경우, 즉 일정하게 턴이 바뀌지 않는 보드게임에는 트리 형태의 알고리즘을 사용하기 어렵고 이러한 경우에는 문제 공간의 크기 또한 급격히 증가한다는 문제로 인해서 트리 기반의 탐색 기법으로 문제를 해결하기가 어려워진다.

이를 해결하기 위해 본 논문에서는 강화 학습(Reinforcement Learning)을 이용한 효율적인 최적해 탐색 기법을 연구하였다. 강화 학습은 주어진 환경에서 어떤 행동을 취하는 것이 가장 효율적일지를 알아내기 위해 각 행동에 따른 보상을 이용하는 기계학습 기법의 하나로, 보상은 양의 값 또는 음의 값을 가질 수도 있다. 이 보상이 누적되어서 가장 큰 보상을 가질 때를 최선의 전략으로 정한다. 강화 학습은 체스[4], 오셀로[5]와 같은 게임 AI분야뿐만 아니라 로봇 제어, 엘리베이터 스케줄링[6] 등 다양한 분야에서 널리 사용되고 있다.

본 연구에서 강화 학습을 선택한 이유는 다른 알고리즘들과 달리 현재 상태, 행동, 보상에만 영향을 받는 강화학습의 특성상 턴이 일정하지 않더라도 잘 동작할 수 있으리라 판단했기 때문이다. 게임을 이기거나 점수를 획득하였을 때 양의 보상을 주고, 게임을 지거나 점수를 잃었을 때 음의 보상을 주는 시행을 반복하여 가능한 모든 보드의 상태에서 어디에 두는 것이 가장 효율적일지를 계산해보았다. 제안한 방법의 성능을 확인하기 위하여 무작위로 행동하는 알고리즘, 점수를 최대한 잃지 않는 탐욕적인 알고리즘, 알파베타 가지치기를 활용한 알고리즘, 미니맥스 알고리즘을 이용한 전략과 비교하여 그 결과를 확인해 보았다.

2. 강화 학습(Reinforcement Learning)

강화 학습이란 잘한 행동에 대해 칭찬받고 잘못한 행동에 대해 벌을 받는 경험을 통해 자신의 지능을 키워나가는 학습법이다. 각 환경에는 목적 달성에 필요하거나 필요하지 않은 다양한 행동들이 존재한다. 만약 목적 달성에 필요한 쪽으로 행동을 취한 경우 양의 보상을 주고, 목적 달성에 불필요한 행동을 취한 경우 음의 보상을 준다. 이를 반복하다보면 목적 달성에 필요한 쪽으로 모델이 학습된다.



[그림 1] 강화 학습의 모식도

강화 학습은 다음 세 가지 구성 요소로 이루어져 있다.

1. 환경 상태 집합 S
2. 행동 집합 A
3. 포상의 집합 R

모든 시점에 에이전트는 자신의 상태 $s_t \in S$와 그 때 취할 수 있는 행동의 집합 $A(s_t)$를 가지고 있다. 에이전트가 어떠한 행동 $a \in A(s_t)$를 실행하면 그에 따라 새로운 환경 s_{t+1}과 행동에 따른 보상 $r(a)$를 받게 된다. 강화 학습 에이전트는 누적된 보상 $R = \Sigma \gamma^t r_{t+1}$이 최대가 되도록 행동한다. 여기서 γ는 미래의 보상이 현재에 비해 얼마나 가치 있는지를 나타내는 할인율(discount factor)로, 일반적으로 0과 1사이의 값을 가진다. 즉, 강화 학습의 목적은 가장 보상이 크게 되는 $\pi : S \rightarrow A$를 찾아내는 것이다. 강화 학습의 의사 코드(pseudo code)는 다음과 같다.

```
function R_learning(state)
    if state is a terminal state
        return end_game
    select an action in this state
    R(state, action)=R(state, action) + Reward(state, action)
    R(state, action)=R(state, action) + gamma*R_learning(next_state)
    //next state is a state after action
    return R(state, action)
end function
```

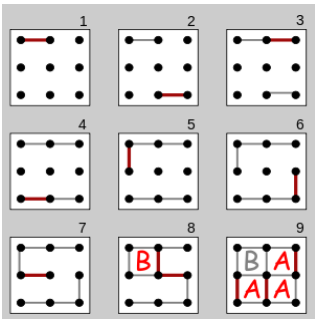
[그림 2] 강화 학습 과정의 의사 코드(pseudo code)

1~2번째 줄에서 현재의 상태가 종결 상태에 해당되는지 판별한다. 3번째 줄에서 현재의 상태에서 가능한 모든 행동 중 하나의 행동을 선택한다. 선택하는 과정은 누적된 보상에 비례하는 룰렛 휠 선택방식(Roulette Wheel Selection)을 채택하였다. 룰렛 휠 선택방식이란 각 해의 품질을 평가하고, 품질에 따라 차등하게 선택될 확률을 조절하는 방식을 뜻한다. 즉, R(state, action)이 클수록 그 행동이 선택될 확률이 높아진다. 여기서 R(state, action)은 현재 상태에서 이 action을 취하였을 때의 누적된 보상을 의미한다. 우선 4번째 줄에서 R(state, action)값에 행동에 따른 보상을 더해준다. 그리고 그 action 후에 있을 보상을 구하기 위해서 R(state, action)에 R_learning(next_state)과 gamma의 곱을 더해준다. 그리고 최종적으로 R(state, action)을 반환함으로써 함수가 종결된다.[4]

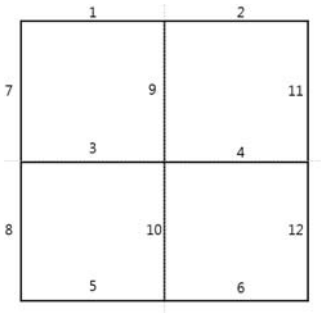
학습을 수차례 반복하다보면 목표 달성에 도움이 되는 행동에 대한 누적 보상은 커지고, 목표 달성에 도움이 되지 않는 행동의 누적 보상은 작아진다. 즉, 점점 목표 달성에 가까운 행동을 선택할 확률이 높아지고, 학습된 모델은 결국 목표 달성에 가까운 행동을 선택할 수 있게 된다.

3. Reinforcement Learning을 이용한 턴이 일정하지 않은 보드게임의 최적해 탐색

본 논문에서는 'Dots and Boxes'라는 게임의 최적해를 강화학습을 이용하여 탐색하였다. 'Dots and Boxes'는 점과 선으로만 진행하는 게임으로 두 플레이어가 번갈아가면서 정사각형 모양의 격자의 점들 사이를 연결하여 정사각형의 네 변이 모두 연결했으면 마지막 선분을 그은 사람에게 득점이 인정된다. [그림 3]에서 보면 7번 상태에서 첫 번째 칸에 세 개의 선분이 그어져 있고, 8번 상태로 넘어가면서 B가 마지막 선분을 그어 B의 득점이 인정됨을 확인할 수 있다. 득점을 하게 되면 해당 정사각형 영역에 표시를 하고, 모든 칸에 선분을 놓게 되면 게임이 완료되고, 더 많은 칸을 차지한 사람이 승리하는 게임이다.



[그림 3] Dots and Boxes 모식도



[그림 4] 보드 기록 방법

Dots and Boxes 게임에 강화 학습을 적용시키기 위하여 모든 보드의 상태를 기록하는 것이 요구되었다. 그래서 그림 4와 같이 모든 선분에 1부터 12까지의 숫자를 부여하였다. 만약 [그림 3]의 2번 상태와 같이 1번과 6번 선분에만 선분이 그어져 있다면 $000000100001_{(2)}$로 보드의 상태를 기록하였다. 이렇게 되면 Dots and Boxes 게임의 종결 상태는 $111111111111_{(2)}$가 된다. 각 상태에 따라 취할 수 있는 행동은 12개의 선분밖에 존재하지 않으므로 [그림 4]와 같이 1~12까지의 숫자로 대응시켜 기록하였다. 또, 누적 보상을 이차원 배열로 설정하여 R[$000000100001_{(2)}$][2]와 같은 형태로 저장하였다. 4*4 배열에 대해서도 같은 방법으로 강화 학습의 결과를 저장하였다.

보상은 득점, 실점, 게임의 승패 유무에 적용하였다. 강화 학습 진행 정도와 성능 면에서 고려해 보았을 때, 득점과 실점은 각각 1점과 -1점의 보상을, 승리와 패배에 각각 5점과 -5점의 보상을 주었다. 그 예로 그림 2에서 강화 학습 에이전트가 B라고 가정하면 7번 상태에서 8번 상태로 넘어갈 때, 1의 보상을 얻게 된다. 즉, $100001110111_{(2)}$ 상태에서 8번 선분을 긋는 경우의 누적 보상 R[$100001110111_{(2)}$][8]에 1을 더해준다.

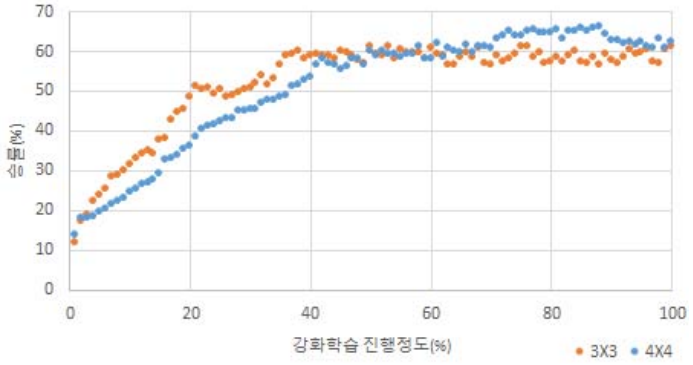
강화학습을 진행하기 위해서는 강화학습으로 학습을 진행하는 전략의 상대가 되어줄 전략이 필요하다. 무작위하게 행동하는 전략, 실점을 최소화하기 위하여 탐욕적으로 행동하는 전략, 알파베타 가지치기를 이용한 전략, 미니맥스 알고리즘을 이용한 전략 총 4종류 전략의 성능 비교를 통해 가장 성능이 좋게 나온 알파베타 가지치기 전략을 학습에 이용하였다. 네 전략 간의 승률은 [표 1]에 나타내었다.

[표 1] AI간의 승률(1000회 시행, 승률은 1열에 쓰여 있는 AI 기준)

>>상대	Random	Greedy	Alpha-Beta Pruning	Minimax
Random		0%	0%	32.3%
Greedy	100%		20%	90%
Alpha-Beta Pruning	100%	80%		95.6%
Minimax	67.6%	10%	4.4%	

4. 실험 결과

Dots and Boxes의 다양한 전략 중 알파베타 가지치기 전략을 상대 전략로 설정하여 강화 학습을 진행하였다. 3X3 격자의 경우 1,000,000번의 게임을 진행하여 10,000번마다 승률을 구해서 강화 학습 정도를 살펴보았고, 4X4 격자의 경우 100,000번 게임을 진행하여 1,000번마다 승률을 구했다. 구한 승률을 그래프로 나타내면 다음과 같다.



[그림 5] 강화 학습의 진행 정도에 따른 승률

[그림 5]에서 보면 3X3 판에 대해서 400만 세대 전에는 꾸준히 증가하는 양상을 보이다가 400만 세대 이후에는 승률이 56%~61% 사이에 진동하는 결과를 얻었다. 또, 4X4 판에 대해서는 5만 세대까지는 꾸준히 증가하는 양상을 보이다가 그 이후에는 60%~65% 사이에서 진동하는 결과를 얻었다.

3X3 판과 4X4 판에서 진행한 강화 학습의 결과를 네 종류의 AI와 대결시켰을 때의 승률은 다음과 같다.

[표 2] 강화 학습 AI의 성능 분석(1000번 시행)

size	Random	Greedy	Alpha-Beta Pruning	Minimax
3X3	100	60.4	58.1	61.5
4X4	100	63.3	63.3	66.6

실험 전 가장 성능이 좋았던 알파베타 가지치기 전략을 상대 전략으로 삼아서 학습을 하였다. 그 결과, 다른 전략에 대해서도 3X3 판에서는 60%, 4X4 판에서는 65%의 승률을 가지는 것을 [표 2]에서 확인할 수 있다. 이를 통해 상대가 알파베타 가지치기인 경우에만 최적화된 전략이 아닌 필승전략에 가깝게 학습이 진행되었음을 알 수 있었다. 또, 3X3보다 4X4가 더 승률이 높게 나온 것은 판의 크기가 커짐에 따라 트리 알고리즘의 탐색 시간과 탐색 공간이 큰 폭으로 증가하여 성능이 상대적으로 좋아진 것으로 추측된다.

5. 결론 및 발전 방향

본 논문에서는 강화 학습 기법을 활용하여 턴이 넘어가지 않는 경우가 있는 게임 Dots and Boxes의 전략을 제작하였다. 강화 학습을 사용한 이유는 현재의 상태, 보상, 행동에만 영향을 받기 때문에 알파베타 가지치기, 미니맥스 알고리즘과 같은 트리 알고리즘보다 턴이 넘어가지 않는 경우에 대한 처리를 손쉽게 할 수 있을 것이라 판단했기 때문이다. 또, 트리 알고리즘의 일종인 알파베타 가지치기, 미니맥스 알고리즘과 탐욕적인 알고리즘, 무작위 알고리즘과의 비교를 통해 강화 학습 기법을 통하여 만들어낸 게임 전략이 어느 정도의 효율을 가지는지 확인하였다. 그 결과, 트리기반의 전략들과 비교하여 상대적으로 높은 승률을 가지는 것을 확인할 수 있었다.

강화 학습은 학습 시간은 오래 걸릴지라도 학습된 데이터만 가지고 있다면 매우 빠르게 게임을 진행할 수 있다는 장점이 있다. 또, 인간 사용자와의 플레이를 통해서도 전략을 발전시킬 수 있다. 마지막으로 강화 학습을 통해 구한 전략이 트리 알고리즘을 이용한 전략보다 좋은 성능을 나타냄을 승률을 통해 확인할 수 있었다. 이는 다른 트리 알고리즘이 턴이 유지되지 않는 경우나 판의 크기 증가에 취약하기 때문이다. 따라서 턴이 바뀌지 않는 게임 전략에 강화 학습이 적합하다는 결론을 내릴 수 있다.

더 높은 성능을 가지는 게임 전략을 제작하기 위해서는 게임 횟수를 늘리거나 한 가지 인공지능이 아닌 다양한 종류의 인공지능을 통해 학습하는 노력이 필요할 것 같다. 또, 더 적합한 Reward를 만든다면 좋은 성능의 게임 전략을 구성할 수 있으리라 생각된다.

참고문헌

[1] Li, Shuqin, Dongming Li, and Xiaohua Yuan. "Research and Implementation of Dots-and-Boxes Game System." *Journal of Software 7.2 (2012)*: p256-p262.

[2] Anthony Knittel, and others. “Stochastic Reinforcement in Evolutionary Multi-Agent Game Playing of Dots-and-Boxes”, International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce, p54, 2006

[3] Anthony Knittel, Terry Bossomaier, and Allan Snyder. “Concept Accessibility as Basis for Evolutionary Reinforcement Learning of Dots and Boxes”, Computational Intelligence and Games, p140~p145, 2007

[4] Block, Marco, et al. "Using reinforcement learning in chess engines." Research in Computing Science 35,p31~p40, 2008

[5] van Eck, Nees Jan, and Michiel van Wezel. "Application of reinforcement learning to the game of Othello." Computers & Operations Research 35.6 ,p1999~p2017, 2008

[6] Barto, A., and R. H. Crites. "Improving elevator performance using reinforcement learning." Advances in neural information processing systems 8, p1017~p1023, 1996