강화학습 수업
정리

2022313107 정성규

Model Free Prediction

Model Free Prediction 정의

- MDP를 모를 때 가치 함수를 예측하는 방법
- Monte-Carlo Learning (MC)
- Temporal Difference Learning (TD)

MDP와 Model Free Prediction의 차이점.

Transition Probability Matrix의 유무가 기본적인 정의이나, 통상적으로, State Size의 크기에 따른 가치함수 연산량에 따라 사용 수단이 다르다. MDP는 결국 수렴할 수 있고, MFP는 수렴하지 못할 수 있기에 일부 위험요소는 있으나 빠르게 결과를 예측할 수 있다는 장점이 있다. 강화학습은 Model-Free 하다. 우리는 MDP 또는 Transition Probability를 모르기 때문에 확정적으로 모델링할 수 없다. (즉, S T A R이 정확하지 않다.)

Monte Carlo Learning

- MC는 시뮬레이션(모의) 결과로부터 직접적으로 학습을 한다.
- MC는 MDP를 몰라도, Historical data를 통해 이를 구현할 수 있다.(Model Free)
- MC는 episode를 완료한 후 이를 학습한다.
- MC는 각 에피소드 상 State의 Gain의 평균을 학습한다. (Gain : sum of response reward) 가치함수와 Q함수를 찾기 위해서는 순차적인 State의 진행이 필요하다. MC는 여러 에피소드의 진행 결과들을 종합하여, 가치함수를 구하는 방법이다. 각 에피소드를 통해 State에 따른 Gain을 구할 수있는데, 그래서 모든 에피소드 내 State에 따른 Gain의 평균을 활용하여 정책과 가치함수를 추론하는 방법을 Monte-Carlo simulation이라 한다.

이 방법의 핵심은 다음 state 이동을 합리적으로 추론하는 것인데, transition probability가 없는 현 상황에서 이를 추론하는 것은 쉽지 않다. 이 때, 우리는 history data를 참고하여 이를 추론한다. 경험적 자료들을 활용하면 여러 에피소드를 만들어낼 수 있고 이 에피소드에 따른 gain들을 구하여 state 이동에 따른 이들의 Reward 평균을 구하여 가치함수와 Q 함수를 업데이트 할 수 있다.

MC에는 두가지 방법이 있다 Every visit MC / First visit MC가 그것이다. 지금까지 설명한 것은 every visit MC이고 FVMC는 각 에피소드 별 처음으로 State에 방문한 Gain만을 활용하여 가치함수를 업데이트 하는 방법이다.

블랙잭 Ex, PPT 10p

블랙잭 설명은 PPT 참조, 12~13이 되면 카드를 더 받을지 덜 받을지를 결정해야 하는 상황인데, 딜러는 무조건 카드를 받아야 하고 게이머는 더 받을지를 결정할 수 있는 권한이 있다. 이 게임은 간단하지만 전략이 필요하다. Transition Probability를 통하여 순간순간 Action을 결정할 수도 있지만, 랜덤하게 여러번 실행한 결과값을 활용하여 next state를 찾아가는 방법이 더욱 간단할 수 있다. 200*200 메트릭스를 계산하는 것보다는 시뮬레이션을 하는 것이 더욱 편하기에, 강화학습을 사용하는 것이 타당할 수 있다. 이 때 사용하는 방법이 몬테카를로 시뮬레이션이다.

MC정책의 한계

일반적으로 MC는 분산이 큰 방법이다. 이를 줄이기 위해서는 많은 데이터가 필요한데 이를 수집하는데 큰 돈이 필요한 경우 이 방법은 적절하지 않다. 그리고 가치함수를 업데이트 하기 위해서는 에피소드가 끝나야 하는데 이 에피소드가 일반적인지 확실하지 않은 점이 있다.

Temporal Difference Learning

- TD는 에피소드로부터 바로 학습을 할 수 있다.
- TD는 MC와 동일하게 Model-Free하다
- TD는 bootstrapping을 통해 끝나지 않은 에피소드를 통해서도 학습할 수 있다.

TD는 벨만 함수에서 다음 state의 가치함수를 현재 가지고 있는 v(st+1)을 가지고 업데이트를 한다. 이는 MC에 비해 초기에 에러 값이 높은 단점이 있다.

MC vs TD

- MC는 분산이 크지만, Bias가 없다.
- TD는 분산은 작으나, Bias가 있다.
- TD는 MDP환경에 MC보다 더 적합하다.
- TD는 부족하지만 현재 갖고 있는 값을 통해 업데이트를 진행한다.

Random Walk Ex, PPT 23p

24p 그래프를 보면 TD가 MC보다 RMS감소가 큰 것을 알 수 있다.
시뮬레이션에서 alpha의 의미는 모의 간 새로운 값의 업데이트하는데 반영할 값의 비율을 말하는 것이다. 이는 딥러닝의 learning rate와 같다.

FVMC와 TD의 차이점

FVMC는 일단 gain을 바탕으로 하기 때문에 기본적으로 여러 스텝 후의 State 상황을 반영하고, 이는 더 빠른 업데이트를 보여주고, TD는 각 State 상에서 다음 State의 가치함수만을 활용하여 업데이트하기 때문에 초기에는 Reward가 있는 가치함수만 업데이트되어 상대적으로 천천히 업데이트 되는 경향이 있다. 하지만 TD는 실제 행한 결과 값을 바탕으로 업데이트하는 반면, MC는 estimated한 미래 예측 값을 기초로 업데이트한다.

Simulation

Discrete Event Simulation – 이벤트가 발생하였을 때 state가 변화하는 모의 환경.
그렇다면 이벤트와 state를 정의해야 함.

Ex – Access Router

1. make an abstract – arrival과 departure event가 있을 것이다. 각 state는 위 Event에 따라 변화할 것이다.
2. 이와 유사한 상황으로 Queueing system 이 있어 여기의 theory를 활용하여 모의를 진행하고 결과를 예측할 수 있다.

3. 모의를 하기 위해서는 내가 풀고 싶은 system의 boundary를 정하고 여기에 맞는 state와 action을 정의해야 한다. 그리고 이에 맞는 event를 정의하고 handle하면 가상 모의 환경이 완성된다.

Single queue model

1. 모델 정의는 PPT 참고
2. 각 event 발생 간격을 정의하기 위해 interarrival time을 정의함. 이 때, 모든 event를 동시에 발생하지 않는다. 전후 시간간격을 꼭 둬야 한다.
3. 이 때 Poisson Process를 따르는 M/M/1 모의는 interarrival time이 exponential을 따른다.
4. 그리고 service time또한 exponential을 따른다.
5. running simulation by hand page를 연산하다 보면 특정 packet이 waiting time이 발생하는 것을 알 수 있다. (arrival time과 service begin간의 갭) 이처럼 손으로도 모의 환경을 연산, 구현할 수 있다. 하지만 시간이 오래 걸린다.
6. Utilization = working time / total time