Ch 9. On-policy Prediction with Approximation

Outline

Intro

- Part 2 시작하는 장으로 함수 근사(Function Approximation)에 대한 내용 포함
- 가중치 벡터 w를 사용하는 함수 V(s,w) 으로 Parameterized 된 형식으로 표현
- Part 1에서 사용하는 Tabular에서 더 큰 state-space 가진 문제에 적용**(Large-Scale Reinforcement Learning)**
- 더 큰 state-space에 대해서는 비용과 데이터가 부족하기에 limited data를 활용하여 근사의 해를 찾아야 함
- Value function에 대해서 일반화하여 function approximation을 하는 것
- 강화학습에서 적용하고자 하는 '일반화'의 개념은 :

  ->Value function의 sample을 근거로 function approximation을 구하기 위한 일반화의 개념

  함수 근사의 예시 : Machine learning, Neural Network, Pattern Recognition, statistical curve fitting, supervised/unsupervised learning
- 본 장에서는 state가 있을 때의 value function approximation에 대해서 논의
- 그 이후 장에서는 control, off-policy, n-step, policy gradient에 대해 논의

1

Value-function Approximation

Value-function Approximation

- Part 1에서는 특정 state의 value-function v(s)을 target으로 reward 주는 방식으로 update함
- 정답이 있는 supervised learning과 비슷한 개념
- 강화학습은 supervised learning처럼 training data-set이 있는 것이 아니라, data가 실시간으로 점진적으로 추가됨
- 강화학습에서의 Target은 지도학습처럼 고정이 아닌 Nonstationary Target Function임
- 그러므로 Non-stationary target을 처리할 수 있는 function approximation이 필요함

2

The Prediction Objective(VE)

Prediction Objective

- Tabular case에서는 target value에 수렴하므로 objective function불필요
- But approximation function을 사용할 경우에는 모든 state에 수렴할 수 없어서 MSE(Mean squared value Error) VE는 다음 식과 같이

$$\overline{\mathrm{VE}}(\mathbf{w}) \doteq \sum_{s \in \mathcal{S}} \mu(s) \Big[v_\pi(s) - \hat{v}(s,\mathbf{w}) \Big]^2 .$$

- 추정치가 실제값과 얼마나 다른지 나타내는 척도(회귀에서 많이 사용)
- $\mu(s)$: Target policy에서 state s가 어느 시간 동안 있었는지 표현으로 on-policy에서는 **On-policy Distribution**으로 표현
- Continuing task의 경우 Target policy에서는 stationary distribution, episodic task일 경우에는 initial state에 따라 분포가 변경 가능

3 Stochastic-gradient and Semi-gradient Methods

Stochastic-gradient and Semi-gradient Methods

- 함수 근사 방법은 종류가 많아, 본 책에서는 Linear Gradient Descent 방법을 주로 설명 예정
- Stochastic Gradient Descent(SGD) : 함수 근사 방법 중 가장 많이 사용되는 방법, online reinforcement에 적합
- Weight vector : 고정된 개수의 real valued component를 가진 column vector임
- SGD에서는 step마다 VE를 줄이는 방향으로 weight vector(w)에 대한 update가 진행됨

$$\begin{aligned}\mathbf{w}_{t+1} &\doteq \mathbf{w}_t - \frac{1}{2}\alpha\nabla\Big[v_\pi(S_t) - \hat{v}(S_t,\mathbf{w}_t)\Big]^2 \\ &= \mathbf{w}_t + \alpha\Big[v_\pi(S_t) - \hat{v}(S_t,\mathbf{w}_t)\Big]\nabla\hat{v}(S_t,\mathbf{w}_t),\end{aligned}$$

- Gradient-descent : 그라디언트(벡터 버전 미분값)을 사용해서 오차를 줄임
- Stochastic : 업데이트마다 샘플이 어떤가에 따라 error가 커지기도 하고 작아지기도 함

Stochastic-gradient and Semi-gradient Methods

- Using SGD in Monte Carlo target Gt

Gradient Monte Carlo Algorithm for Estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, S_1, A_1, \ldots, R_T, S_T$ using π
  Loop for each step of episode, $t = 0, 1, \ldots, T-1$:
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[G_t - \hat{v}(S_t, \mathbf{w})\big]\nabla \hat{v}(S_t, \mathbf{w})$

- Gradient의 일부만 포함하기 때문에 이 방법은 Semi-gradient method라고 부름
- Gradient method만큼 안정적으로 수렴되지는 않지만, 선형 함수 등에는 수렴값이 양호함
- Semi의 경우에는 속도가 빠르고, 에피소드가 끝날 때까지 기다리기보다는 online continual하도록 연속적인 업데이특 가능함

Stochastic-gradient and Semi-gradient Methods

- Semi-gradient method 원형은 semi-gradient TD(0)임

Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \to \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose $A \sim \pi(\cdot|S)$
    Take action A, observe R, S'
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})\big]\nabla\hat{v}(S,\mathbf{w})$
    $S \leftarrow S'$
  until S is terminal

- State aggregtation : 함수 근사를 일반화하는 간단한 형태인데, 각 그룹에 대해 하나의 추정값(가중치 벡터 w의 한 구성 요소)를 사용하여 상태를 그룹화함
- 상태의 가치는 해당 그룹의 구성요소로 추정함

4 Linear Methods

Linear Methods

- Function approximation에서 가장 특별한 case는 weight vector의 linear function case라고 함

$$\hat{v}(s,\mathbf{w}) \doteq \mathbf{w}^\top \mathbf{x}(s) \doteq \sum_{i=1}^{d} w_i x_i(s).$$

- State,s에 대해 weight,w와 동일한 차원인 실수가 있고, w와 x(s)를 내적하여 state value를 approximation하는 방법임
- X(s)는 feature vector라고 부름 => 근사 함수들의 집합에 대한 linear basis를 형성하기에 basis function이라고 부름.
- 본 방법을 사용하면 SGD update가 간단해지므로, 가장 선호되며 실제로 많이 사용됨
- Linear이기에 수렴을 보장하며, 계산측면에서 효율적임
- Local Optimum에 수렴이 보장만 된다면, Global Optimum 수렴이 보장됨

5

Feature Construction for Linear Methods

Feature Construction for Linear Methods

- Linear Method가 다양한 linear feature 적용 case

1. 다항식(Polynomials)

- Interpolation 및 Regression과 공통점이 많음
- Feature를 다항으로 표현하는 것은 성능은 좋지 않지만 가장 심플하고 기초적인 방법이며 관련 내용을 소개하는 데 있어서 좋은 출발점임
- State를 나타내는 숫자가 2개인 경우에서 k개인 경우로 일반화해서, 문제의 state가 갖는 차원들 사이의 복잡한 상호작용을 표현할 수 있음
- 높은 차수의 다항식은 더 복잡한 함수에 대한 정확한 approximation이 가능하게 됨

Feature Construction for Linear Methods

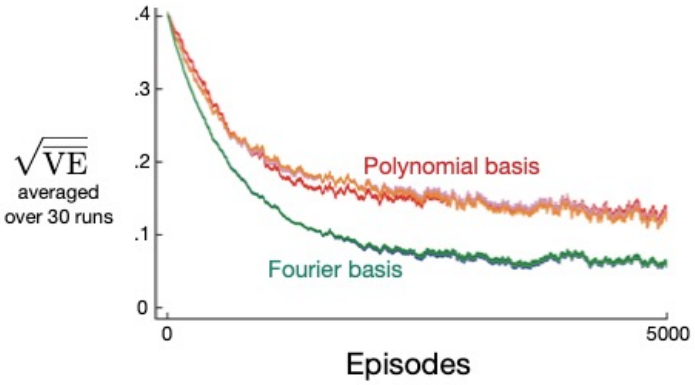
2. 푸리에(Fourier Basis)

- 서로 다른 진동수를 갖는 사인과 코사인의 weighted sum으로 주기 함수를 표현기반으로 함
- 주기를 갖는 사인 및 코사인 함수의 linear combination으로 대상 함수를 포현
- 주기성을 갖는 사인 및 코사인 특징의 linear combination에서 한 주기에 해당하는 함수가 타겟 대상이 됨
- Semi-gradient TD(0) or SARSA 학습 알고리즘에 사용시에는 시간 간격 파라미터를 각 특징에 대해 서로 다른 값을 적용하면 도움이 됨
- SARSA에 푸리에 코사인을 적용하면 다른 여러 가지 feature function과 비교해보았을 때 좋은 성능을 낼 수 있음

Feature Construction for Linear Methods

2. 푸리에(Fourier Basis)

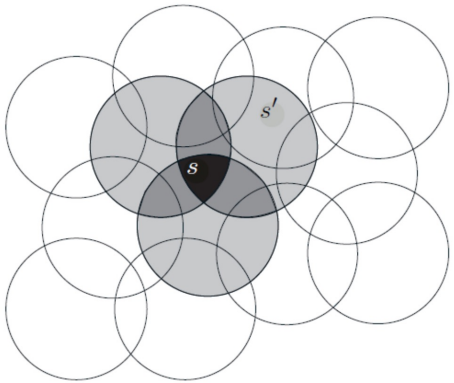
- Online learning에서는 다항식을 사용하는 것을 권유하지 않음
- 1000개의 상태를 가진 random walk에서 푸리에와 다항식 basis를 비교하는 그래프



Feature Construction for Linear Methods

3. 엉성한 부호화(Coarse Coding)

- State-set이 2차원인 경우에는 state가 2차원에서 한 점으로 나타나고, 2개의 실수 성분을 가진 vector가 됨

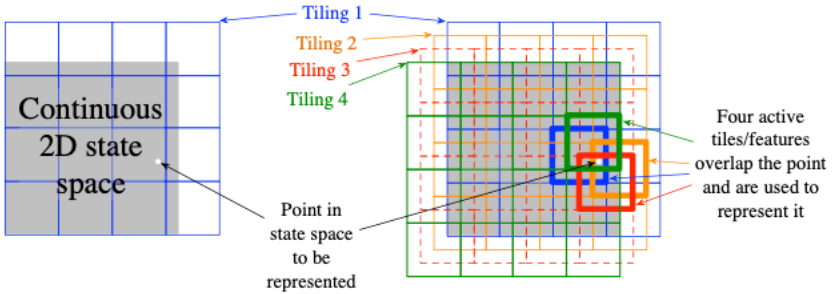


- 우측 그림에서 만약 상태 s가 해당 원 안에 있으면 1, 원 밖에 있으면 0의 값을 가지는 binary feature 방법을 사용하는 방식을 'coarse coding'이라고 함
- 원의 크기가 작으면 각 업데이트가 여러 feature에 일반화 되는 것이 적고, 크기가 크면 많음
- 일반화가 많이 된다는 의미는 training example에 대해서 더 많은 다른 state들에 weight vector가 영향을 미치게 된다는 것임
- 즉 겹치는 Feature가 있는 상태를 나타낸 것을 Coarse coding이라고 부른다고 정리

Feature Construction for Linear Methods

4. 타일 부호화(Tile Coding)

- Tile coding은 coarse coding의 한 종류임

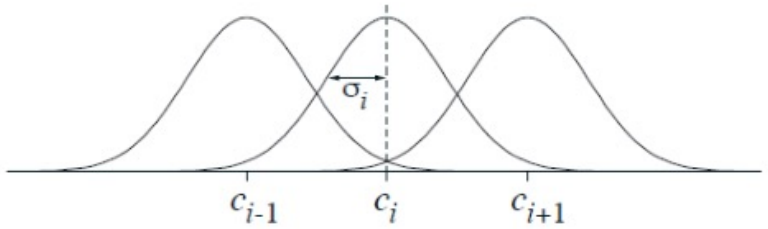


- Sequential digital computer를 위한 가장 실용적인 표현법 중 하나
- 특징의 수용 영역이 여러 개의 부분으로 나누어진 state-space 각 부분으로 묶이며, 이 부분을 '타일 영역'이라고 부름
- 어떤 state에 대해 특정 시점에 feature의 개수가 모든 State에 대해 동일하다는 장점이 있음
- 메모리 요구사항을 줄이는 'hashing이라는 트릭이 있는데, 이는 차원의 저주를 극복할 수 있는 방법임

Feature Construction for Linear Methods

5. 방사형 기저 함수(Radial Basis Function)

- RBF(Radial Basis Function)은 coarse coding의 0 혹은 1의 binary feature를 일반화하여 0과 1사이의 실수값을 가지게 함



- RBF에서 자주 쓰이는 형식이 가우시안 형태인데, 여기서 c는 센터, a는 feature의 넓이임
- Binary feature에 비해서 부드럽게 변하고 미분 가능한 근사 함수를 도출함
- 그러나 현업에서는 크게 고려하지는 않고, 계사량이 많이 필요하며 수동 tuning이 필요함

6 Nonlinear Function Approximation : Artificial Neural Networks

Nonlinear Function Approximation : Artificial Neural Networks

- 인공신경망은 비선형 함수를 근사할 때 많이 사용
- 인공신경망의 학습 목적은 가중치에 대한 최적의 값을 결정하는 것
- 역전파(backpropagation) 알고리즘을 사용
- 강화학습에서는 Deep convolution neural network을 성공적인 유형이라고 함

7 Summary

Summary

- 강화학습 시스템이 인공지능 규모나 큰 문제에 적용되려면 일반화 능력이 필요
- Supervised learning function approximation을 위한 다양한 방법이 제안
- 지도학습 방법의 한 종류인 Parameterized function approximation을 활용하는 방법
- 가중치 벡터를 찾기 위한 방법으로 가장 좋은 방법인 SGD(Stochastic Gradient Descent)
- 인공신경망에서는 심층강화학습(Deep Reinforcement learning)을 사용

Thank You

Reference

- https://irealist.org/data-science/?mod=document&uid=7952&pageid=1
- https://keepmind.net/%EA%B0%95%ED%99%94%ED%95%99%EC%8A%B5-9-on-policy-prediction-with-approximation/
- https://jay.tech.blog/2017/01/02/on-policy-prediction-with-approximation/
- https://talkingaboutme.tistory.com/entry/RL-The-Objective-for-On-policy-Prediction