Off-Policy Methods with Approximation

준경사도 방법

- 비활성 정책 방법은 갱신 목표를 변경하는 것이 어렵다.
- Semi-gradient Expected Sarsa

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\delta_t \nabla \hat{q}(S_t, A_t, \mathbf{w}_t), \text{ with} \tag{11.5}$$

$$\delta_t \doteq R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})\hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t), \text{ or} \quad \text{(episodic)}$$

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \sum_a \pi(a|S_{t+1})\hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t). \quad \text{(continuing)}$$

- 표 기반의 경우, 유일한 행동이 있지만, 함수 근사의 경우 근사에 기여하는 서로 다른 상태-행동 쌍에 서로 다른 가중치를 부여하고자 한다.

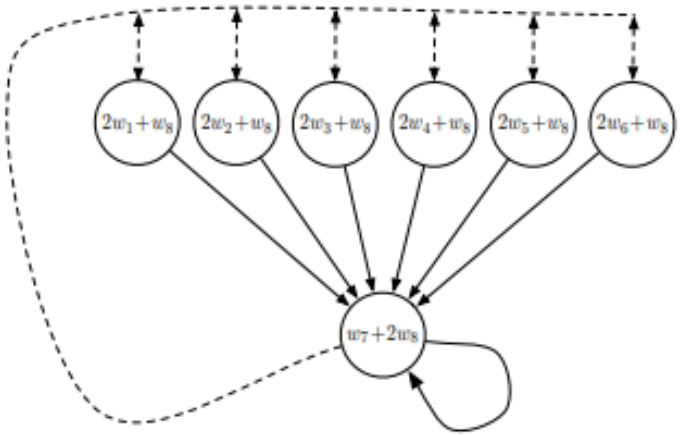
비활성 정책 발산의 예제

추정값 w 의 발산

W가 갱신되지 않은 채로 전이가 반복해서 발생

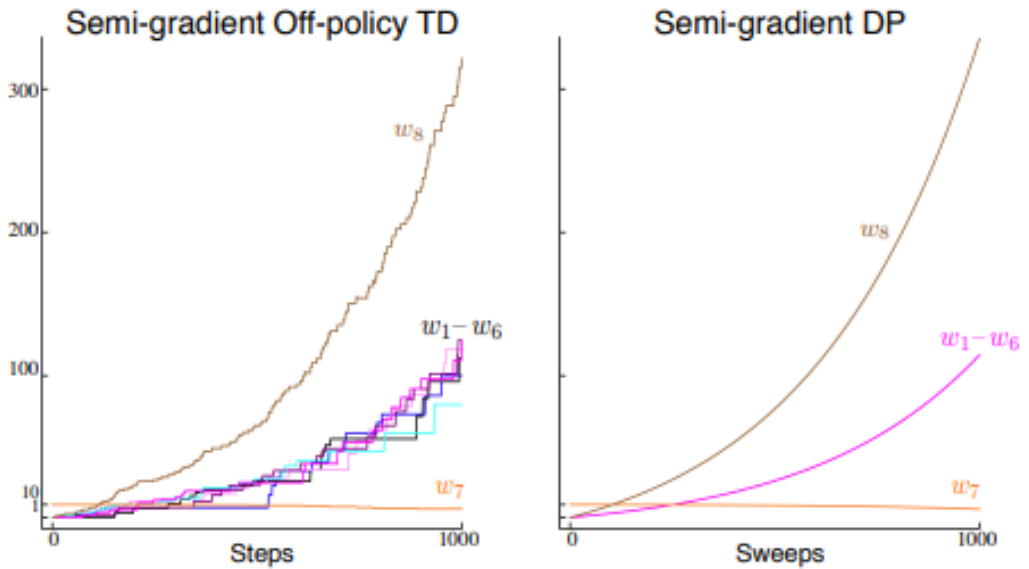
비활성 정책은 미래 보상에 대한 약속이 주어지고 나면, 목표 정책이 선택하지 않을 행동을 취한 이후에, 약속이 이뤄지지 않아도 되는 상황 발생

Baird's counterexample



불안정성은 모든 양의 시간 간격에 대해 발생한다.
동적 프로그래밍(DP)에서도 발생한다.

$$\mathbf{w}_{k+1} \doteq \mathbf{w}_k + \frac{\alpha}{|\mathcal{S}|} \sum_s \Big(\mathbb{E}_\pi[R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}_k) \mid S_t = s] - \hat{v}(s, \mathbf{w}_k) \Big) \nabla \hat{v}(s, \mathbf{w}_k).$$



치명적인 삼위일체

불안정성 및 발산의 위험 발생

Function approximation

피해갈 수 없다.

메모리와 컴퓨터의 계산 능력을 능가하는 상태공간

데이터의 양이 많아질수록 복잡도가 너무 많아져 비용이 많이 든다.

Bootstrapping

없이 할 수 있지만, 계산 효율성은 포기해야한다.

계산량과 메모리를 절약

상태로 돌아왔을 때 상태를 식별할 수 있는 능력을 활용하여 학습이 수행되기 때문에 부트 스트랩을 하면 종종 더 빠른 학습이 가능하다. 상태가 잘 표현되지 않으면 발생하는 편차가 오차를 더 크게 발생시킨다.

Off-policy training

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

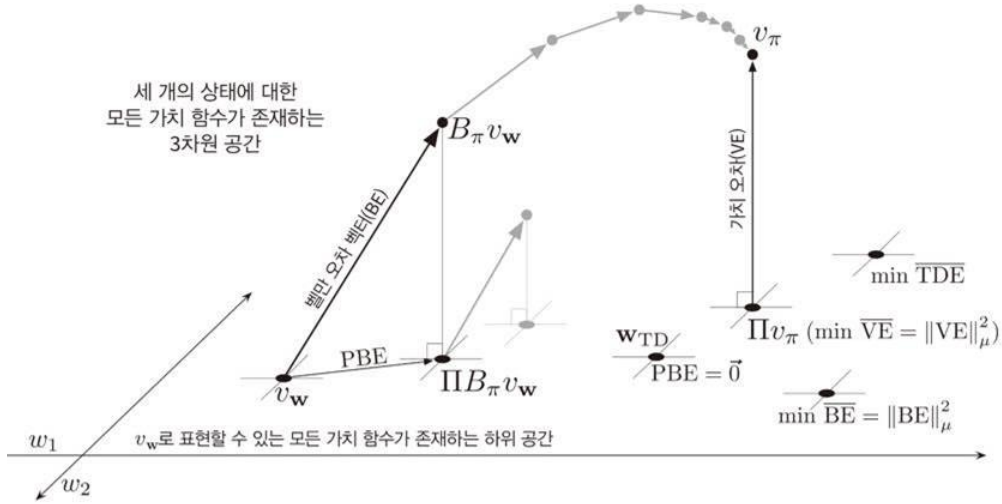
선형 가치 함수 기하 구조

- 세개의 상태, 두개의 파라미터
- 하나의 함수로 근사할 수 없는 정책
- 거리를 측정해야한다

$$\|v\|_\mu^2 \doteq \sum_{s\in\mathcal{S}} \mu(s) v(s)^2.$$

벨만 방정식. 벨만 작용자

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right], \qquad \text{for all } s \in \mathcal{S}.$$



선형함수 근사를 이용하면 평균 제곱 투영 벨만 오차(PBE)를 0으로 만드는 근사적 가치 함수가 항상 존재한다. 가치함수는 가치오차, 벨만 오차를 최소화하는 가치함수로의 수렴성을 보장한다. (11.7,8)

벨만 오차에서의 경사도 강하

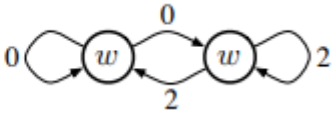
- 잔차 경사도 (Naive Residual-gradient) 알고리즘
  다음 상태로의 전이가 결정론적이거나 시뮬레이션에서 다음 상태에 대한 독립적인 표본을 얻을 때 가능하다.
  벨만 오차를 줄이는 것이 바람직한 목표가 안될 수 있다. 엉뚱한 해를 도출한다.

$$\begin{aligned}\bar{\delta}_{\mathbf{w}}(s) &\doteq \left(\sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_{\mathbf{w}}(s')\right]\right) - v_{\mathbf{w}}(s) \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma v_{\mathbf{w}}(S_{t+1}) - v_{\mathbf{w}}(S_t) \mid S_t = s, A_t \sim \pi\right],\end{aligned}$$

$$\begin{aligned}\overline{\text{TDE}}(\mathbf{w}) &= \sum_{s \in \mathcal{S}} \mu(s) \mathbb{E}\left[\delta_t^2 \mid S_t = s, A_t \sim \pi\right] \\ &= \sum_{s \in \mathcal{S}} \mu(s) \mathbb{E}\left[\rho_t \delta_t^2 \mid S_t = s, A_t \sim b\right] \\ &= \mathbb{E}_b\left[\rho_t \delta_t^2\right]. \qquad \text{(if } \mu \text{ is the distribution encountered under } b\text{)}\end{aligned}$$

벨만 오차는 학습할 수 없다.

- 학습가능하다 : 잘 정의 되어있고 환경의 내부 구조에 대한 정보가 주어져서 계산할 수 있다.
- 모든 상태가 하나의 성분으로 이루어진 동일한 특징 벡터 x=1을 갖고 가치의 근삿값 w를 갖는다.



- VE가 다르지만 생성되는 데이터는 동일한 분포를 갖기 때문에, 학습될 수 없다.
- VE를 학습할 수 없지만, VE를 최적화하는 파라미터를 학습할 수 있다!
- 평균제곱이득오차(Mean Square Return Error, RE)

$$\begin{aligned}\overline{\mathrm{RE}}(\mathbf{w}) &= \mathbb{E}\left[\left(G_t - \hat{v}(S_t,\mathbf{w})\right)^2\right] \\ &= \overline{\mathrm{VE}}(\mathbf{w}) + \mathbb{E}\left[\left(G_t - v_\pi(S_t)\right)^2\right].\end{aligned}$$

Thanks!