* One-Period MDP

- 정의
  - $N=2$ 로, $T=\{1,2\}$ 임.
  - State space S는 유한하며, action set A_s는 모든 $s\in S$에 대해서 유한함.
  - $g_1(s,a)$: 첫번째 decision epoch을 지나고 난 뒤 얻는 보상
  - $g_2(s')$: 마지막 단계인 s'에서 얻게 되는 마지막 종결의 보상.
  - 목적: $\max_a \{ g_1(s,a) + E[g_2(s')] \}$.
  - $\pi = (d_1)$ $(a' = d_1(s))$ policy를 실행하고 첫번째 state가 s로 주어질 때 전체 보상의 기댓값은 다음과 같음.

$$V(s,a') = g_1(s,a') + E_s^{\pi}[g_2(X_2)]$$

$$= g_1(s,a') + \sum_{s'\in S} \underbrace{P_1(s'|s,a)}_{\substack{t=1 \text{ 일때 } s \text{ 가 주어질 때} \\ t=2 \text{ 에서 } s' \text{가 될 확률}}} g_2(s')$$

  - $$V^*(s,a_s^*) = \max_{a'\in A_s} V(s,a')$$

$$= \max_{a'\in S} \{ g_1(s,a') + \sum_{s'\in S} P_1(s'|s,a) g_2(s') \}$$

    - 각각의 state $s\in S$에 대해서, 전체 보상의 기댓값 $V(s,a')$를 최대화하는 action $a_s^* \in A_s$를 찾는 것이 목표.
  - S와 A_s가 유한하므로, 이를 최대화시키는 액션 a^*가 하나 이상 존재.

* One-Period MDP - Randomized.

- 초기 state는 s이고, $q(a)$라는 확률로 액션 $a\in A_s$를 실행할 때 (randomized), 전체 보상의 기댓값은 다음과 같음.

$$E_q[V(s,\cdot)] = \sum_{a\in A_s} q(a)\cdot V(s,a) = \sum_{a\in A_s} q(a) [g_1(s,a) + \sum_{s'\in S} P_1(s'|s,a) g_2(s')]$$

  - $\sum_{a\in A_s} q(a) = 1$ 이며, $q(a) \geq 0$ 임.
- $\max_{q\in P(A_s)} \sum_{a\in A_s} q(a) V(s,a) = \max_{a'\in A_s} V(s,a')$ 이므로, randomized π도 deterministic π 만큼 최고의 결과를 낼 수 있음.

* Two-State MDP

- Notations
  - $T = \{1, 2, \dots, N\}$, $N \leq \infty$
  - $S = \{s_1, s_2\}$, $A_{s_1} = \{a_{11}, a_{12}\}$ $A_{s_2} = \{a_{21}\}$.
- 보상과 전이 확률은 다음과 같음.
  - $g_t(s_1, a_{11}) = 5$, $g_t(s_1, a_{12}) = 10$, $g_t(s_2, a_{21}) = -1$

    $g_N(s_1) = 0$, $g_N(s_2) = 1$
  - $P_t(s_1 \mid s_1, a_{11}) = 0.5$, $P_t(s_2 \mid s_1, a_{11}) = 0.5$

    $P_t(s_1 \mid s_1, a_{12}) = 0$, $P_t(s_2 \mid s_1, a_{12}) = 1$

    $P_t(s_1 \mid s_2, a_{21}) = 0$, $P_t(s_2 \mid s_2, a_{21}) = 1$

* Single Product Stochastic Inventory Control

- 전체적인 메커니즘.

a_t만큼 order

D_t 계속 주문 받음

t S_t ——— $t+1$ S_{t+1}

손실 : $o(a_t) + h(S_t + a_t)$

이득 : $F(u)$

$$= \sum_{j=0}^{S_t + a_t} P_j f(j) + q_{S_t + a_t} \cdot f(S_t + a_t)$$

D_t 충족

$$S_{t+1} = \max(S_t + a_t - D_t, 0)$$
$$= [S_t + a_t - D_t]^+$$

S_t : t 시점에서 재고 상황

a_t : 공급되는 제품 수

D_t : 수요, 임의적임.

백로깅 없는 경우, $D_t - S_t - a_t$는 전부 버림.

* MDP 공식

- t 시점에서 보상의 기댓값 : $g_t(S_t, a_t, S_{t+1}) = f(S_t + a_t - S_{t+1}) - o(a_t) - h(S_t + a_t)$
  - $o(u)$: u 만큼 주문하는데 드는 비용. $(\underbrace{K}_{고정} + \underbrace{c(u)}_{변동})$
  - $h(u)$: 한 달 동안 u 단위 유지비용.
  - $r(u)$: 월 말에 남은 u 단위의 가치
  - $f(j)$: j 만큼 팔아서 얻는 수익.
- 이 때 $f(S_t + a_t - S_{t+1})$은 t 시기의 수익이므로 S_{t+1}을 모름. ⇒ $F(u)$로 대체
- $F(u) = \sum_{j=0}^{u-1} P_j f(j) + q_u f(u)$
  - P_j : t 시점에 j 만큼 수요가 발생할 확률
  - q_u : $\sum_{j=u}^{\infty} P_j = P(D_t \geq u)$, 수요가 u를 초과할 확률
- 따라서, t 시점의 보상의 기댓값은 다음과 같음.
  - $g_t(s, a) = F(s+a) - o(a) - h(s+a)$

    $g_N(s) = r(s)$

* 전이 확률

- s에서 a를 추가해 s'가 될 확률은 다음과 같이 표현.

$$P_t(s'|s,a) = \begin{cases} 0 & \text{if } M \ge s' > s+a \\ P_{s+a-s'} & \text{if } M \ge s+a > s' > 0 \\ q_{s+a} & \text{if } M \ge s+a,\ s'=0 \end{cases}$$

* 예제

- Q1) 만약 주문을 넣은 후에 수요 D_t가 들어오면 MDP를 어떻게 수정해야 하는가?
  - A1) Q : 목표 재고량, Q-σ : 최소 충족
    - 만약 월 초에 재고량이 σ수준 이하면 Q-s 만큼의 주문량을 넣음. 만약 재고가 σ 단위 이상이면 주문량은 교체하지 않음.
    - 따라서 고정 정책은 $d(s) = \begin{cases} 0 & s \ge \sigma \\ Q-s & s < \sigma \end{cases}$ 가 됨.
  - Q2) 만약 백오깅이 허용되면, 즉 수요가 들어온 후 주문한 재고가 들어오면?
  - A2) $S_t \in S = \{\dots,-2,-1,0,1,2,\dots\}$ 이며 음수에 백오깅함.
    - 따라서 h(u)를 u<0에 대해서도 적용
    - u<0인 경우에는 demand를 즉각처리하지 못하였기 때문에 패널티 비용 과부
      - $h(u) = \begin{cases} -\phi u & (u<0) \\ \varphi u & (u>0) \end{cases}$

* Optimal Stopping problems.

- 설명.
  - uncontrolled Markov Chain
    - 전이 확률이 action의 함수가 아님. State 마다 2개의 action 존재
      - ↳ Continue & Stop
    - 만약 t시기에 $\begin{cases} \text{Stop 하면 } r_t(s) \text{ 만큼의 보상.} \\ \text{Continue 하면 } f_t(s) \text{ 만큼의 비용.} \end{cases}$
    - finite process 이며, t=N 시기에 h(s) 만큼의 보상 얻음.
    - 한번 Stop 하면 더 이상의 진행은 X.

* MDP 공식

- 보상 함수는 다음과 같음.

$$g_t(s,a) = \begin{cases} -f_t(s) & \text{if } s \in S,\ a = C \\ +r_t(s) & \text{if } s \in S,\ a = Q \text{ or } s = s' = \Delta,\ a = C \\ 0 & \text{if } s = \Delta \end{cases}$$

Q. absorbing state

$$g_N(s) = \begin{cases} h(s) & \text{if } s \in S' \\ 0 & \text{if } s = \Delta \end{cases}$$

A. 어떤 상태에 도달한 이후 거기서 빠져나오지 못하는 것을 의미.

- 전이 확률은 다음과 같음.
  - $$P_t(s'|s,a) = \begin{cases} P_t(s'|s) & \text{if } s', s \in S,\ a = C \\ 1 & \text{if } s \in S,\ s' = \Delta,\ a = Q \\ & \quad \text{or} \\ & \quad s = s' = \Delta,\ a = C \\ 0 & \text{otherwise} \end{cases}$$
  - 전이 확률은 중단 시점에서의 $r_t(s)$와 $\sum_{t=0}^{t-1} f_t(s)$ 간의 차이 최대화를 목적으로 진행

* Controlled Discrete-time Dynamic Systems

- 설명.
  - 전이 확률 대신에 sample path와 system equation을 사용해서 설명.
  - $S_{t+1} = f_t(S_t, a_t, w_t)$ 이며, $w_t \in W$ 이고 w_t는 t시점에서의 랜덤 방해 변수
  - $S_t \in S$ 시퀀스들은 통제 하의 변수들 $\{a_1, a_2, ..\}$와 통제 외 변수들 $\{w_1, w_2, ..\}$로 방해 받음.
  - w_t는 다른 t의 w_t와 독립적이며, $g_t(\cdot)$는 W의 PDF이고 이는 s, a와 독립임.
  - t시기에 보상 $g_t(s_t, a_t)$를 받으며, 마지막 $t=N$일 때는 $g_N(s_N)$의 보상 받음.

* MDP formulation

- $t<N$ 일 때 $g_t(s_t, a_t)$, $t=N$일 때 $g_N(s_N)$
- $P_t(s'|s,a) = p(s' = f_t(s,a,w_t)) = \sum_{w \in W,\ s' = f_t(s,a,w)} g_t(w)$

* MDP와 Controlled discrete-time Dynamic System의 차이

- MDP는 s_t, a_t로 인해 정의되는 전이확률로 인해 시스템이 운행됨.
- 반면, Controlled discrete-time Dynamic System은 $s_{t+1} = f_t(s_t, a_t, w_t)$로 정의된 state를 따르는 전이확률이 교란변수 w_t의 분포 $q_t(w)$에 의해 도출됨.

* Economic Growth Model

- 자본의 투자나 소비에 대한 계획 경제에 대한 확률적 역할 모델.
- t 시기의 자본 s_t을 관찰하여 얼만큼 소비할 지 a_t를 결정하고, $s_t - a_t$를 투자에 사용함.
- 소비하자마자 즉시 효용 $\varphi_t(a_t) = g_t(s, a)$가 만들어지며, $s_{t+1} = w_t F_t(s_t - a_t)$가 됨.
  - F_t : 현재 남은 자본에 대한 기대 수익, w_t : 교란변수
- 전이 확률은 다음과 같음
  - $P_t(s' | s, a) = P(s' = w_t F_t(s-a)) = \sum_{w \in W, s' = wF_t(s-a)} q_t(w)$