Lecture D2. Markov Reward Process 2

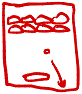
Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

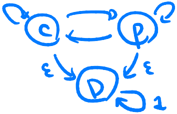
SEOULTECH 서울과학기술대학교 데이터사이언스학과

I. Motivation

Recap

- A Markov chain is a stochastic process with the specification of
  - a state space S
  - a transition probability matrix $\mathbf{P}$
- A Markov reward process is a Markov chain with the specification of
  - a reward r_t with the reward function $R(s)$
  - a time horizon H, which is the duration we are interested in cumulative sum of rewards.
- If H is finite, then we call *finite-horizon MRP*.
- IF H is infinite, then we call *infinite-horizon MRP*.
  - Sometimes, the stochastic process is *non-terminating infinite horizon MRP*.
  - Sometimes, the stochastic process is *terminating infinite horizon MRP*. In this case, we may treat them as non-terminating, but the chain is absorbed into a absorbing state whose reward is zero.





Formulating an infinite horizon MRP

- In the previous lecture, we dealt with the following question.

  Given I drink coke today, what is likely my consumption for upcoming 10 days? *(Pepsi is \$1 and Coke is \$1.5)*

- Infinite horizon problem is such as following.

  I am to live eternally. Given I drink coke today, what is likely my consumption for my upcoming forever life? *(Pepsi is \$1 and Coke is \$1.5)*

  (In this case, how to model her death?)

- It may seem unrealistic on this soda problem to have an infinite time horizon. But infinite horizon model is indeed more common for MRP due to following reasons.

1. Time horizon may be finite, but uncertain how long.
2. The horizon may be belived to be a long time.
3. In accounting principle, all businesses are assumed to be perpetual.
4. Oftentimes, each time step is very small such as minute ,or even millisecond, making the number of total time step as a very large number.
5. Really long finite time horizon can be approximated by infinite time.

Return for infinite horizon

- Return for finite horizon in the previous lecture was

$$G_t = \sum_{i=t}^{H-1} r_i = r_t + r_{t+1} + \cdots + r_{H-1}$$

- If extended for infinite horizon, it becomes

$$G_t = \sum_{i=t}^{\infty} r_i = r_t + r_{t+1} + \cdots$$

Convergence of G_t

$$G_t = \sum_{i=t}^{\infty} r_i = r_t + r_{t+1} + \cdots$$

- Is G_t a convergent series, thus measurable??
  - Even if r_i is a small number, it may diverge unless r_t decays drastically over time.
  - What does it mean *drastically*?
    - cf) $\boxed{\sum 1/n = \infty}$ and $\boxed{\sum 1/n^2 < \infty}$ (참고)

Discount factor

(prev) G_t : sum of remaining reward

(now) G_t : discounted sum of remaining reward

Introducing discount factor

- A mathematically convenient way to guarantee is to introduce *discount factor*, $\gamma < 1$ (gamma)
- Using a discount factor, the return, G_t, is newly defined as

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots$$

- As long as r_t is bounded, i.e. $|r_t| < M$ for some $M > 0$ and for all t, G_t is convergent.
- G_t can be written as

$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$$

$G_t = r_t + \gamma r_{t+1} + \cdots$

$< M + \gamma M + \gamma^2 M = \frac{M}{1-\gamma}$

$\therefore$ G_t is convergent

Note that this generalizes the previous notation with $\gamma = 1$.

Is discount factor practical?

- Many real problems indeed should be modelled with discount factor.
- Humans behave in much the same way, putting more importance in the near future.
- Interest rate is generally positive, making today's money worth more than tomorrow's money.
- Future is risky to some degree, making future's reward less valuable than today's reward.
- ~~If you die today, there is no tomorrow.~~

State-value function

- Like before, the state-value function $V_t(s)$ for a MRP and a state s is defined as the expected return starting from state s at time t, namely,

$r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$

$$V_t(s) = \mathbb{E}[G_t | S_t = s]$$

- For infinite horizon problem, are the following two quantity different?
  1. $V_t(s) = \mathbb{E}[G_t | S_t = s]$
  2. $V_0(s) = \mathbb{E}[G_0 | S_0 = s]$

- It is not! This is because the lengths of remaing time where returns are summed are equally infinite.
- This makes our life easier, and allowing us to drop the time subscript for the state-value function when necessary.
- Namely, $V_t(s) = V_0(s) = V(s)$.

```
"Dream as if you'll live forever. Live as if you'll die today. - James Dean"
```

Summary

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots \quad (1)$$

$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i \quad (2)$$

$$V(s) = \mathbb{E}[G_t | S_t = s] \quad (3)$$

$V_{\cancel{t}}(s) = V_{\cancel{0}}(s) = V(s)$

II. Method 3 - Analytic solution

Development

$V_0(s)$, $V_1(s)$, $\cdots$ $V_H(s)$

- For a finite horizon MRP, the goal was to find $V_t(s)$ for all states s for $0 \leq t \leq H$.
- Since $V_0(s) = V_t(s) = V(s)$, the goal is only to find $V(s)$ for all states s.

$$\begin{aligned}
V(s) &= V_t(s) = \mathbb{E}[G_t | S_t = s] \\
&= \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | S_t = s] \\
&= \mathbb{E}[r_t | S_t = s] + \gamma \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s] \\
&= R(s) + \gamma \mathbb{E}[G_{t+1} | S_t = s] \\
&= R(s) + \gamma \sum_{\forall s'} \mathbb{P}[S_{t+1} = s' | S_t = s] \mathbb{E}[G_{t+1} | S_t = s, S_{t+1} = s'] \\
&= R(s) + \gamma \sum_{\forall s'} \mathbf{P}_{ss'} \mathbb{E}[G_{t+1} | S_{t+1} = s'] \ (\because \text{ Markov property}) \\
&= R(s) + \gamma \sum_{\forall s'} \mathbf{P}_{ss'} V_{t+1}(s') \\
&= R(s) + \gamma \sum_{\forall s'} \mathbf{P}_{ss'} V(s') \qquad (4)
\end{aligned}$$

지난번과 같음

(Bellman's Eq)

- The section for `Method 2 - Iterative solution` in the previous lecture had the following equation in `D1.p23`.

$$V_t(s) = R(s) + \overset{\gamma=1}{\sum_{\forall s'}} \mathbf{P}_{ss'} V_{t+1}(s'), \ \forall s$$

V (finite horizon)

- The Eq (4) in the previous page was

$$V(s) = R(s) + \gamma \sum_{\forall s'} \mathbf{P}_{ss'} V(s'), \ \forall s$$

V (infinite horizon)

- Difference
  - The former had $\gamma = 1$.
  - The latter dropped the time subscripts.
- Similarity
  - Both are understood as

$$\left(\text{Expected return at time } t\right) = \left(\text{reward at time } t\right) + \left(\text{Expected return at time } t+1\right)$$

- The above equation is called *Bellman's equation*, named after Richard R. Bellman `(wiki link)` who introduced dynamic programming in 1953.

Analytic formula

$$V(s) = R(s) + \gamma \sum_{\forall s'} \mathbf{P}_{ss'} V(s'), \ \forall s$$

c p c p r P c p

- Once again, the strategy is
  - Column vector v for $V(s)$
  - Column vector R for $R(s)$
  - $\gamma \mathbf{P} v$ for $\gamma \sum_{\forall s'} \mathbf{P}_{ss'} V(s')$, where $\mathbf{P}$ is a transition matrix
- It follows $v = R + \gamma \mathbf{P} v$
- This can be solved as:

$$\begin{aligned} & v = R + \gamma \mathbf{P} v \\ \Rightarrow \quad & Iv = R + \gamma \mathbf{P} v \\ \Rightarrow \quad & Iv - \gamma \mathbf{P} v = R \\ \Rightarrow \quad & (I - \gamma \mathbf{P}) v = R \\ \Rightarrow \quad & v = (I - \gamma \mathbf{P})^{-1} R \end{aligned}$$

Example

I am to live eternally. Given I drink coke today, what is likely my consumption for `my upcoming forever life`*? (Pepsi is \$1 and Coke is \$1.5)*

- We need information regarding the discount rate. Let's assume $\gamma = 0.95$.
- (equivalent to having daily interest rate of 5%)
- We have

$$
\begin{aligned}
v &= R + \gamma \mathbf{P} v \\
\begin{pmatrix} v(c) \\ v(p) \end{pmatrix} &= \begin{pmatrix} R(c) \\ R(p) \end{pmatrix} + \gamma \begin{pmatrix} \mathbf{P}_{cc} & \mathbf{P}_{cp} \\ \mathbf{P}_{pc} & \mathbf{P}_{pp} \end{pmatrix} \begin{pmatrix} v(c) \\ v(p) \end{pmatrix} \\
\begin{pmatrix} v(c) \\ v(p) \end{pmatrix} &= \begin{pmatrix} 1.5 \\ 1.0 \end{pmatrix} + 0.95 \begin{pmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} v(c) \\ v(p) \end{pmatrix}
\end{aligned}
\tag{5}
$$

```r
P <- array(c(0.7,0.5,0.3,0.5), dim=c(2,2))
R <- array(c(1.5,1.0), dim=c(2,1))
gamma = .95
v <- solve(diag(2)-gamma*P)%*%R # v=(I-gamma P)^{-1}R
v
```

```
##          [,1]
## [1,] 26.48148
## [2,] 25.86420
```

$\gamma^{365} = 0.97$

Exercise 1

What is the relationship between the above vector v and stationary distribution? Find an equality.

$$\frac{5}{8} \times 1.5 + \frac{3}{8} \times 1.0 \approx 1.3$$

$$1.3 + 0.95 \times 1.3 + \cdots$$

$$= \frac{1.3}{1-0.95} = 26$$

Exercise 2

What are your concerns for this approach? (Hint: inverting the matrix)

1. Invertible? (mathematically)
2. Invertible? (computationally)

III. Method 4 - Iterative solution - by fixed point theorem

Recap

- The previous approach was based on the following two formula.

$$\underline{v = R + \gamma \mathbf{P} v} \qquad \text{Bellman} \tag{6}$$

$$\underline{v = (I - \gamma \mathbf{P})^{-1} R} \tag{7}$$

- The Eq. (6) is a Bellman's equation.
- The Eq. (7) is used to find a analytic solution.
- Using the Eq (7), there are two concerns that you should have. (This is the suggested solution to Exercise 2)
    1. The matrix $I - \gamma \mathbf{P}$ may not be invertible.
    2. Even if invertible, it may be prohibitive if for a big matrix.
- (1) We are free from the first concern. The matrix $I - \gamma \mathbf{P}$ can be proved to be invertible always as long as $\mathbf{P}$ is stochastic.
- (2) We are not free from the second concern. So, this section introduces an alternative, numerical, and iterative approach.

Iterative algorithm

- Using the fixed-point theorem along with Eq. (6), we apply the following iterative algorithm to find v. (Warning: The subscript i is not state index, not time index, but the iteration index)

$V = R + \gamma P V$

$$v_{i+1} \leftarrow R + \gamma \mathbf{P} v_i$$

$V = 0$

$V \leftarrow R + \gamma P V$

```
Let epsilon <- 10^{-8} # or some small number
Let v_0 <- zero vector
Let v_1 <- R + \gamma*P*v_0
i <- 1
While ||v_i-v_{i-1}|| > epsilon # may use any norm
  v_{i+1} <- R + \gamma*P*v_{i}
  i <- i+1
Return v_{i+1}
```

Math Review - Norm

(0,1) vs (1,3)

1) $\boxed{1+2} = 3$

2) $\sqrt{1^2+2^2}$

3) max(1,2) = 2

Definition 1

(1, 2)

For a length-n vector x, the norm of vector $||x||_p$ is defined as follows.

- 1-norm: $||x||_1 = \sum_{i=1}^{n} |x_i|$ *(sum of absolute value)*
- 2-norm: $||x||_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$ *(Euclidean distance, distance from the origin)*
- ∞-norm: $||x||_\infty = max_{1 \leq i \leq n} |x_i|$ *(farthest axis)*

- For the rest of this course, we will use ∞-norm to guarantee that value functions (or any other quantities) are well approximated for every state.

Implementation

- The pseudo code

```
Let epsilon <- 10^{-8} # or some small number
Let v_0 <- zero vector
i <- 1
While ||v_i-v_{i-1}|| > epsilon # may use any n
  v_{i+1} <- R + \gamma*P*v_{i}
  i <- i+1
Return v_{i+1}
```

- The R-code

```
R <- array(c(1.5,1.0), dim=c(2,1))
P <- array(c(0.7,0.5,0.3,0.5), dim=c(2,2))
gamma <- 0.95
epsilon <- 10^(-8)
v_old <- array(rep(0,2), dim=c(2,1))
repeat{
  v_new <- R + gamma*P%*%v_old
  if (max(abs(v_new-v_old)) < epsilon){
    break
  }
  v_old <- v_new
}
print(v_new)
```

do while

```
##          [,1]
## [1,] 26.48148
## [2,] 25.86420
```

- The full iteration process

```r
R <- array(c(1.5,1.0), dim=c(2,1))
P <- array(c(0.7,0.5,0.3,0.5), dim=c(2,2))
gamma <- 0.95
epsilon <- 10^(-8)
v_old <- array(rep(0,2), dim=c(2,1))
results <- t(v_old)                 # to save
repeat{
  v_new <- R + gamma*P%*%v_old
  if (max(abs(v_new-v_old)) < epsilon){
    break
  }
  results <- rbind(results, t(v_new)) # to save
  v_old <- v_new
}
```

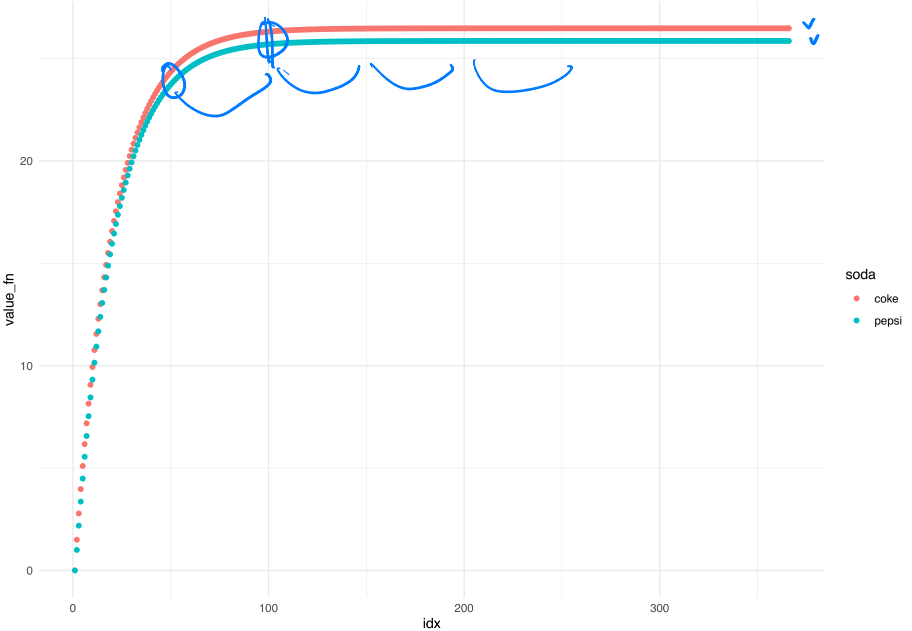
```r
results <- data.frame(results)
colnames(results) <- c("coke", "pepsi")
```

```r
head(results)
```

```
##       coke    pepsi
## 1 0.000000 0.000000
## 2 1.500000 1.000000
## 3 2.782500 2.187500
## 4 3.973800 3.360750
## 5 5.100391 4.483911
## 6 6.169675 5.552543
```

```r
tail(results)
```

```
##         coke   pepsi
## 361 26.48148 25.8642
## 362 26.48148 25.8642
## 363 26.48148 25.8642
## 364 26.48148 25.8642
## 365 26.48148 25.8642
## 366 26.48148 25.8642
```


value_fn
20
10
0
0
100
200
300
idx
soda
coke
pepsi

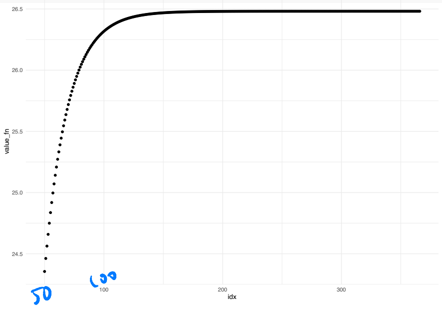
- The previous plot was generated by the following code.

```
library(tidyverse)
results$idx <- as.numeric(row.names(results))
results <- results %>%
  gather("coke", "pepsi", key="soda", value="value_fn")
ggplot(results, aes(x=idx, y=value_fn, group = soda, color = soda)) +
  geom_point() +
  theme_minimal()
```

- Note that there are quite convergence going on after many steps.

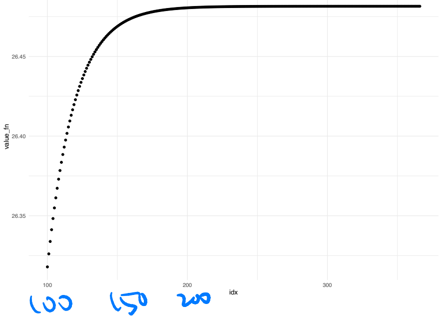
- After 50 steps (coke only)

```
results %>% filter(idx >= 50, soda == "coke") %>%
  ggplot(aes(x=idx, y=value_fn)) +
  geom_point() +
  theme_minimal()
```



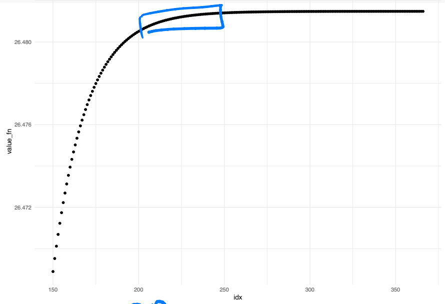
- After 100 steps (coke only)

```
results %>% filter(idx >= 100, soda == "coke") %>%
  ggplot(aes(x=idx, y=value_fn)) +
  geom_point() +
  theme_minimal()
```



- After 150 steps (coke only)

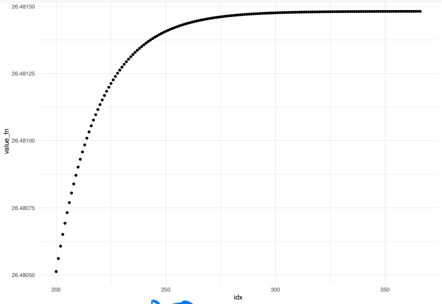
```
results %>% filter(idx >= 150, soda == "coke") %>%
  ggplot(aes(x=idx, y=value_fn)) +
  geom_point() +
  theme_minimal()
```



150 200

- After 200 steps (coke only)

```
results %>% filter(idx >= 200, soda == "coke") %>%
  ggplot(aes(x=idx, y=value_fn)) +
  geom_point() +
  theme_minimal()
```



200 250

QnA

- Q. The iterative algorithm starts with the initial value function estimate of zero. Why does it ever work?
  - A. It works because the meaningful information of reward is repeatedly fed into the iteration process.
- Q. What would be proper value of γ? Would there be an *optimal* value of γ?
  - A. γ is to describe the preference or setting in the real-world.

```
"Success isn't permarnent, and failure isn't fatal. - Mike Ditka"
```