



INDR 450/550

Spring 2022

Lecture 25: Dynamic
Programming (3)

May 23, 2022

Fikri Karaesmen

Announcements

- Class Exercise at the end of lecture today. If you are participating online, please upload your document under Course Contents/Class Exercises
- Project long description due this week
- HW 4 due date May 23 (model reduction, trees, forests etc.)
- **Please complete online course evaluation**

The Problem

- Introduction to Stochastic Dynamic Programming in the context of Capacity Allocation with Multi-class Demand
- Optimal policy by recursion
- Extracting the optimal actions: optimal policy

Policy Evaluation to Optimization

Let $v_1(x)$ be the maximum expected revenue with one period remaining and x items available , we can write

$$\begin{aligned} v_1(x) = & q_1 \max \{ (p_1 + v_0(x-1)), v_0(x) \} \\ & + q_2 \max \{ (p_2 + v_0(x-1)), v_0(x) \} \\ & + q_3 v_0(x) \end{aligned}$$

Recall that $v_0(x) = 0$ for all $x \geq 0$. We can therefore extract the optimal action with 1 period to go: $a(x, 1) = (A, A)$. It is optimal to sell to both classes with 1 period to go.

Policy Optimization

- But now we can do the same for $v_2(x)$.
 - For $0 \leq x \leq Q$ We have:

$$v_2(x) = q_{12} \max \{p_1 + v_1(x-1), v_1(x)\} + q_{22} \max \{p_2 + v_1(x-1), v_1(x)\} + \dots \\ + q_{k2} \max \{p_k + v_1(x-1), v_1(x)\} + q_{02} v_1(x)$$

- Going backwards, for t periods remaining, we have for $0 \leq x \leq Q$:

$$v_t(x) = q_{1t} \max \{p_1 + v_{t-1}(x-1), v_{t-1}(x)\} + q_{2t} \max \{p_2 + v_{t-1}(x-1), v_{t-1}(x)\} + \dots \\ + q_{kt} \max \{p_k + v_{t-1}(x-1), v_{t-1}(x)\} + q_{0t} v_{t-1}(x)$$

- This can be computed if $v_{t-1}(x)$ has already been computed for all x .

Policy Optimization

- Equivalently we can write the optimality conditions as:

$a_1 = A$ if $p_1 \geq v_{t-1}(x) - v_{t-1}(x-1)$, and $a_1 = R$; otherwise

$a_2 = A$ if $p_{2+} \geq v_{t-1}(x) - v_{t-1}(x-1)$, and $a_2 = R$; otherwise

- We note that $\Delta(x) = v_{t-1}(x) - v_{t-1}(x-1)$ is a critical quantity. It corresponds to the marginal value of a seat.
- In the context of capacity allocation $\Delta(x)$ is known as the bid price at time t of seat x .
- It is optimal to sell to a class i customer only if $p_i >$ current bid price.

Policy Optimization

- Obtain $v_t(x)$ by recursion

A	B	C	D	E	F	G	H	I	J	K	L
q1	q2	p1	p2	q3							
0.2	0.7	500	100	0.1							
					186.88	289.296	398.4704	454.8293	493.0157		
v(x,t)											
x↓ t→	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	170	236	288.8	331.04	364.832	391.8656	413.4925	430.794	444.6352	455.7081
2	0	170	340	419.2	493.12	560.704	621.5296	675.5968	723.1759	764.6995	800.6867
3	0	170	340	510	598.28	677.248	753.9392	827.4573	897.0852	962.3033	1022.783
4	0	170	340	510	680	776.452	857.1684	936.5226	1014.71	1091.185	1165.408
5	0	170	340	510	680	850	953.8068	1036.832	1116.77	1196.358	1275.323
6	0	170	340	510	680	850	1020	1130.426	1216.192	1296.712	1376.642
7	0	170	340	510	680	850	1020	1190	1306.384	1395.211	1476.562
8	0	170	340	510	680	850	1020	1190	1360	1481.745	1573.864
9	0	170	340	510	680	850	1020	1190	1360	1530	1656.571
10	0	170	340	510	680	850	1020	1190	1360	1530	1700
11	1	170.1	340.01	510.001	680.0001	850	1020	1190	1360	1530	1700
12	2	171.1	340.2	510.029	680.0038	850.0005	1020	1190	1360	1530	1700
13	3	172.1	341.2	510.3	680.0561	850.009	1020.001	1190	1360	1530	1700
14	4	173.1	342.2	511.3	680.4	850.0905	1020.017	1190.003	1360	1530	1700
15	5	174.1	343.2	512.3	681.4	850.5	1020.131	1190.029	1360.005	1530.001	1700
16	6	175.1	344.2	513.3	682.4	851.5	1020.6	1190.178	1360.044	1530.009	1700.002
17	7	176.1	345.2	514.3	683.4	852.5	1021.6	1190.7	1360.23	1530.062	1700.015
18	8	177.1	346.2	515.3	684.4	853.5	1022.6	1191.7	1360.8	1530.287	1700.085
19	9	178.1	347.2	516.3	685.4	854.5	1023.6	1192.7	1361.8	1530.9	1700.349
20	10	179.1	348.2	517.3	686.4	855.5	1024.6	1193.7	1362.8	1531.9	1701

What is the expected optimal profit (when using the optimal admission policy) if there are 5 seats remaining and 10 periods until the time of flight?

Policy Optimization

- Extract the optimal policy from $v_t(x)$

A	B	C	D	E	F	G	H	I
q1	q2	p1	p2	q3				
0.2	0.7	500	100	0.1				
					186.88	289.296	398.4704	454.8293
$v(x,t)$								
$x \downarrow t \rightarrow$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	170	236	288.8	331.04	364.832	391.8656	413.4925
2	0	170	340	419.2	493.12	560.704	621.5296	675.5968
3	0	170	340	510	598.28	677.248	753.9392	827.4573
4	0	170	340	510	680	776.452	857.1684	936.5226
5	0	170	340	510	680	850	953.8068	1036.832
6	0	170	340	510	680	850	1020	1130.426
7	0	170	340	510	680	850	1020	1190
8	0	170	340	510	680	850	1020	1190
9	0	170	340	510	680	850	1020	1190
10	0	170	340	510	680	850	1020	1190

$a(x,t):$	admit decision for class 2	1 corresponds to admit, 0 corresponds to reject																		
$x \downarrow t \rightarrow$	0	1	2	3	4	5	6	7	8	9										
0	0	0	0	0	0	0	0	0	0	0										
1	0	1	0	0	0	0	0	0	0	0										
2	0	1	1	0	0	0	0	0	0	0										
3	0	1	1	1	1	0	0	0	0	0										
4	0	1	1	1	1	1	1	1	0	0										
5	0	1	1	1	1	1	1	1	1	0										
6	0	1	1	1	1	1	1	1	1	1										
7	0	1	1	1	1	1	1	1	1	1										
8	0	1	1	1	1	1	1	1	1	1										
9	0	1	1	1	1	1	1	1	1	1										
10	0	1	1	1	1	1	1	1	1	1										

If we have two periods remaining and 1 seat available, it is optimal to reject a sale from class 2
 $p_2 + v_1(0) = 100 + 0 \leq v_1(2) = 170$.

Multiple Fare Classes: Bid Prices

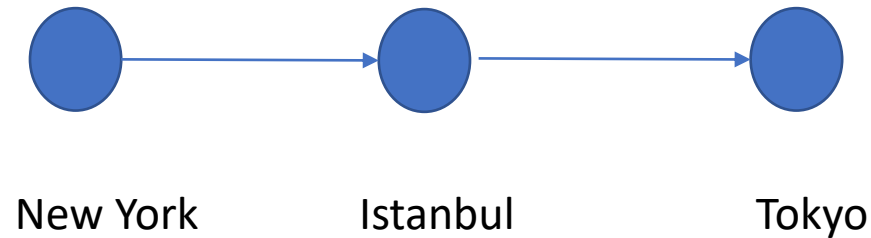
- Some properties:
- $v_t(x)$ is non-decreasing in x . We cannot have a lower expected revenue if there are more seats available.
- $v_t(x)$ is non-decreasing in t . We cannot have a lower expected revenue if there is more time until the end of the flight.
- $v_t(x)$ is concave in x . The expected marginal revenue of an additional seat is non-increasing.
 - The optimal admission rule is a threshold rule (admit if the number of seats available is above a threshold)
 - Class 1 customers are always admitted
- The admission decision in period t depends on: $b_t(x) = v_{t-1}(x) - v_{t-1}(x-1)$. Note that by the above property: $b_t(x) \geq 0$. and by concavity $b_t(x)$ is non-increasing in x .
- $b_t(x)$ is called the bid price with t periods left with x seats available. It defines the admission rule completely.
- If $p_j > b_t(x)$ it is optimal to accept the request from class j , otherwise it is optimal to reject the request.

Multiple Fare Classes: Bid Prices

- If $p_j > b_t(x)$ it is optimal to accept the request from class j , otherwise it is optimal to reject the request.
- If at a given time t , class j requests are rejected, then requests from classes $j+1, j+2, \dots, k$ are also rejected.
- If at a given time t , class j requests are accepted, then requests from classes $1, 2, \dots, j-1$ are also rejected.

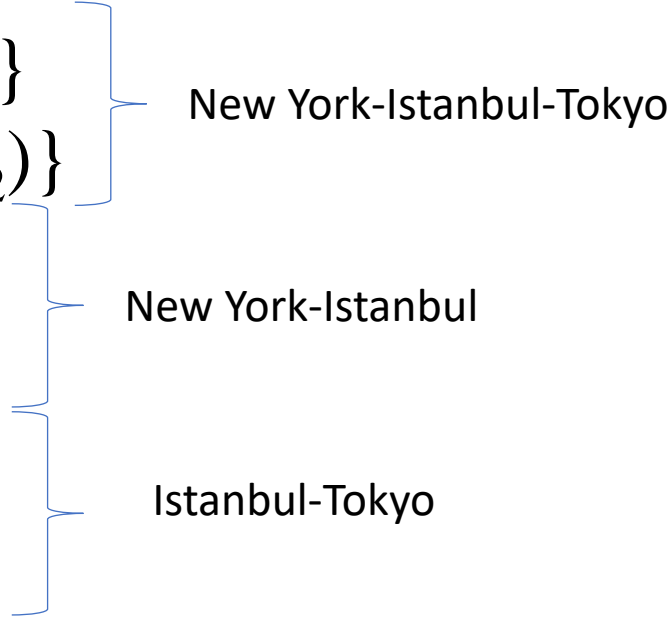
The Need for Approximations

- Consider the same problem but in a small network consisting of two flights:



- There are now passengers flying from New York to Istanbul, from Istanbul to Tokyo and also from New York to Tokyo with a connection in Istanbul.
- Let x_1 be the remaining seats available on the flight from NY to Istanbul
- and x_2 be the remaining seats available on the flight from Istanbul to Tokyo

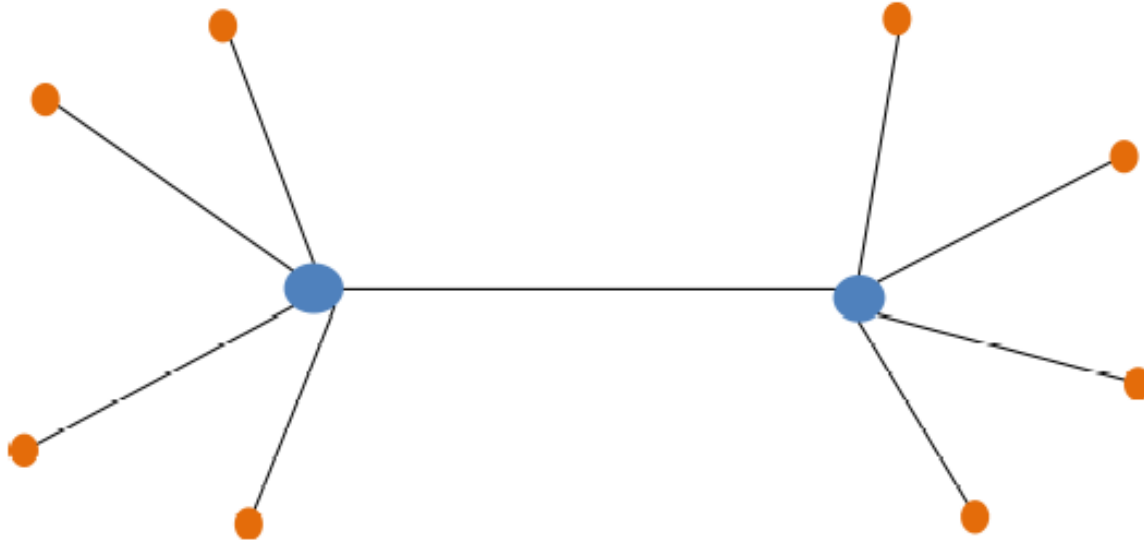
The Network DP

- We can also write and solve a stochastic DP formulation under similar assumptions as before (at most 1 arrival per period etc.)
- We can recursively compute $v_t(x_1, x_2)$ the expected optimal revenue with t periods to go and x_1 and x_2 seats remaining on flights (resources) 1 and 2.
- $$v_t(x_1, x_2) = q_1 \max\{p_1 + v_{t-1}(x_1 - 1, x_2 - 1), v_{t-1}(x_1, x_2)\} + q_2 \max\{p_2 + v_{t-1}(x_1 - 1, x_2 - 1), v_{t-1}(x_1, x_2)\} + q_3 \max\{p_3 + v_{t-1}(x_1 - 1, x_2), v_{t-1}(x_1, x_2)\} + q_4 \max\{p_4 + v_{t-1}(x_1 - 1, x_2), v_{t-1}(x_1, x_2)\} + q_5 \max\{p_5 + v_{t-1}(x_1, x_2 - 1), v_{t-1}(x_1, x_2)\} + q_6 \max\{p_6 + v_{t-1}(x_1, x_2 - 1), v_{t-1}(x_1, x_2)\} + q_0 v_{t-1}(x_1, x_2)$$


This can still be solved numerically but is more difficult. The state space now is in the order of Q^2 .

A more realistic network

- In 1960s major airlines began to establish **hub-and-spoke** networks.



- Arrivals and departures at the hubs are timed so that passengers arriving from cities to the west of the hub can connect to cities to the east, and vice versa.
- The hub-and-spoke network allows an airline to offer a large number of products with a relatively small number of flights. In this example, with 18 flights offers 90 products.

State space is in the order of Q^{18} !

DP Approximations

- One big motivation for an approximation is then the fact that the state space of the DP may be large and the recursion cannot be solved.
- Known as Curse of Dimensionality
- But there are other reasons for approximations:
 - The parameters that feed the model (arrival rates) are estimated from data.
 - These will change over time.
- In particular, the learning issue: estimate parameters and follow an approximately policy at the same time.
 - If all parameters are known, the model is completely specified. Then we can use the backward recursion but when parameters change, we have to look forward. Unfortunately, there is no forward recursion to solve the dynamic optimization problem.

Some Ideas from Reinforcement Learning

- Local policy improvement: if we have approximations under some reasonable policy of the future expected revenue ($v_{t-1}(x)$). We can take a locally (for the current time) optimal action.
 - For instance, the arrival rates (probabilities) may depend on some predictors and are updated at each t .
-
- The current estimates are: \hat{q}_{1t} and \hat{q}_{2t} .
 - Assume also that we have some data corresponding to similar situations which generated future revenues of $\hat{w}_{t-1}(x)$.
 - We can achieve a local improvement for the optimal action at time t by considering:

$$\max \{p_2 + \hat{w}_{t-1}(x - 1), \hat{w}_{t-1}(x)\}$$

Some Ideas from Reinforcement Learning

- Local policy improvement: Multi-step look ahead
- We can possibly get more accurate estimators of the value function if we look ahead by more than one period.
- The optimal action at time $t - 1$ is given by:

$$\max \{p_2 + \hat{w}_{t-2}(x - 1), \hat{w}_{t-2}(x)\}$$

We can now get a more accurate (or updated) estimate for $\hat{w}_{t-1}(x)$.

$$\begin{aligned}\hat{w}'_{t-1}(x) &= \hat{q}_{1,t-1} \max \{p_1 + \hat{w}_{t-2}(x - 1), \hat{w}_{t-2}(x)\} \\ &\quad + \hat{q}_{2,t-1} \max \{p_2 + \hat{w}_{t-2}(x - 1), \hat{w}_{t-2}(x)\} \\ &\quad + \hat{q}_{3,t-1} \hat{w}_{t-2}(x)\end{aligned}$$

Some Ideas from Reinforcement Learning

- Local policy improvement: Multi-step look ahead
 - And eventually possibly a better optimal action for time t
 - The optimal action at time t is now given by:

$$\max \{ p_2 + \hat{w}'_{t-1}(x - 1), \hat{w}'_{t-1}(x) \}$$

- We can also look multiple periods ahead and therefore combine the more accurate short term forecasts with more rough cut long term approximations of the value function.

Some Ideas from Reinforcement Learning

- If we can observe multiple sample paths of the process and the rewards generated, we can approximate an initial value function.
- We can then improve the policy for the next run and obtain better estimates to the value function.
- We can then keep on estimating the value function, improving the policy, estimating the value function, improving the policy...
- ϵ -greedy algorithm: does this but not to get trapped in local optima, also takes some suboptimal actions with a small probability ϵ .
- If the 'games' are easily repeated (by simulation for instance), this gives us a general tool. This is not always easy to do in operations practice.

Some Ideas from Reinforcement Learning

- Approximating the tail of the expected revenue function: $v_{t-1}(x)$.
- Approximating the tail of the profit function is critical to run a data-dependent dynamic policy. One idea is to use a deterministic approximation.
- Assume that with t periods to go and x seats remaining our point estimates for the total demand are \hat{d}_1 and \hat{d}_2 .
- We can argue that the optimal dynamic policy will find a smart way to satisfy all of the demand from class 1 if possible and some of the demand from class 2 if capacity remains.

Some Ideas from Reinforcement Learning

- Approximating the tail of the expected revenue function:

- Here's a deterministic approximation:

$$\tilde{v}_t(x) \approx p_1 \min\{x, \hat{d}_1\} + p_2 \min\{(x - \hat{d}_1)^+, \hat{d}_2\}$$

- We can now look forward using the current estimates of future demand available. As before the optimal action would be given by:

$$\max \{p_2 + \tilde{v}_{t-1}(x - 1), \tilde{v}_{t-1}(x)\}$$

Some Ideas from Reinforcement Learning

- Here's the approximation:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	q1	q2	p1	p2	q3								
2	0.2	0.7	500	100	0.1								
3													
4						160	250	340	400	400			
5	$v(x,t)$												
6	$x \downarrow t \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	
7	0	0	0	0	0	0	0	0	0	0	0	0	
8	1	0	170	260	340	420	500	500	500	500	500	500	
9	2	0	170	340	440	520	600	680	760	920	920	1000	
10	3	0	170	340	510	620	700	780	860	1020	1020	1100	
11	4	0	170	340	510	680	800	880	960	1120	1120	1200	
12	5	0	170	340	510	680	850	980	1060	1220	1220	1300	
13	6	0	170	340	510	680	850	1020	1160	1320	1320	1400	
14	7	0	170	340	510	680	850	1020	1190	1420	1420	1500	
15	8	0	170	340	510	680	850	1020	1190	1520	1520	1600	
16	9	0	170	340	510	680	850	1020	1190	1530	1530	1700	
17	10	0	170	340	510	680	850	1020	1190	1530	1530	1700	
18	11	0	170	340	510	680	850	1020	1190	1530	1530	1700	
19	12	0	170	340	510	680	850	1020	1190	1530	1530	1700	
20	13	0	170	340	510	680	850	1020	1190	1530	1530	1700	
21	14	0	170	340	510	680	850	1020	1190	1530	1530	1700	
22	15	0	170	340	510	680	850	1020	1190	1530	1530	1700	
23	16	0	170	340	510	680	850	1020	1190	1530	1530	1700	
24	17	0	170	340	510	680	850	1020	1190	1530	1530	1700	
25	18	0	170	340	510	680	850	1020	1190	1530	1530	1700	
26	19	0	170	340	510	680	850	1020	1190	1530	1530	1700	
27	20	0	170	340	510	680	850	1020	1190	1530	1530	1700	