SYSEN 5200: Systems Analysis Behavior and Optimization

Dynamic Programming

Nick Kunz [NetID: nhk37] nhk37@cornell.edu

April 30, 2023

1. Suppose that you have N tickets available for a particular concert. On day t, a customer arrives and declares that she is willing to pay P_t dollars for the ticket. You must decide to sell the ticket for P_t or to decline the sale. Suppose that as the date of the concert nears, customers are willing to pay a higher cost for the ticket, so that the distribution of P_t is sampled according to:

$$P(P_t = p) = \begin{cases} \frac{t}{2T} & \text{for } p = 20 \\ \frac{T-t}{2T} & \text{for } p = 10 \\ \frac{1}{2} & \text{for } p = 5 \end{cases} \tag{1}$$

The concert is on the evening of day T, thus any tickets not sold by the end of day T will go to waste. Let $V(t,n)$ denote the expected profit from an optimal ticket sales strategy after (and including) day t if you have n tickets left at the beginning of day t. Using dynamic programming to calculate $V(1,N)$, we compute the optimal strategy for managing ticket sales.

(a) To compute the expected sales profit for the base cases $V(T+1,n)$, we have:

$$V(T+1,n) = 0 \,\forall\, n \in [0,N] \tag{2}$$

(b) The recursive formula for computing $V(t,n)$ is:

$$V(t,n) = \max\left(\sum_p P(P_t = p) \cdot \max(p + V(t+1,n-1), V(t+1,n)), V(t+1,n)\right) \tag{3}$$

(c) The pseudo-code for the dynamic program can be written as:

Algorithm 1 Optimal Ticket Sales

Init: $V = (T+1) \times (N+1)$
for $t = T$ **down to** 1 **do**
  for $n = 0$ **to** N **do**
    not sell $\leftarrow V[t+1][n]$
    sell $\leftarrow 0$
    for $p \in 5, 10, 20$ **do**
      if $n > 0$ **then**
        $P_t \leftarrow 0$
        if $p = 5$ **then**
          $P_t \leftarrow \frac{1}{2}$
        else if $p = 10$ **then**
          $P_t \leftarrow \frac{T-t}{2T}$
        else if $p = 20$ **then**
          $P_t \leftarrow \frac{t}{2T}$
        end if
        sell $\leftarrow$ sell $+ P_t \cdot \max(p + V[t+1][n-1], V[t+1][n])$
      end if
    end for
    $V[t][n] \leftarrow \max(\text{sell}, \text{not sell})$
  end for
end for
Output: $V[1][N] = V(1, N)$

(d) Solving for the setting $N = 3$ and $T = 10$, we have:

```python
## optimal ticket sales
def optim_ticket_sales(N, T):

    ## init
    V = [[0 for i in range(N + 1)] for i in range(T + 2)]

    ## recursion
    for t in range(T, 0, -1):
        for n in range(N + 1):
            not_sell = V[t + 1][n]
            sell = 0
            for p in [5, 10, 20]:
                if n > 0:
                    if p == 5:
                        P_t = 1 / 2
                    elif p == 10:
                        P_t = (T - t) / (2 * T)
                    elif p == 20:
                        P_t = t / (2 * T)
                    sell += (
                        P_t * max(p + V[t + 1][n - 1], V[t + 1][n])
                    )
            V[t][n] = max(sell, not_sell)
    return V, V[1][N]

## results
N = 3
T = 10
V, profit = optim_ticket_sales(N, T)
print("Expected Profit: $", round(profit, 2))
```

Fig. Python 1(d)

Expected Profit: **$50.71**

(e) A table with the values computed for $V(t, n)$ of all values t from 1 to 10, as well as n from 0 to 3, is such that:

```python
## table of values
def table_values(V, N, T):
    column = "t\\n | "
    for n in range(N + 1):
        column += f"{n} | "
    print(column)
    print("-" * len(column))
    for t in range(1, T + 1):
        row = f"{t} | "
        for n in range(N + 1):
            row += f"{V[t][n]:.2f} | "
        print(row)

## results
N = 3
T = 10
V, profit = optim_ticket_sales(N, T)
table_values(V, N, T)
```

Fig. Python 1(e)

$V(t,n)$	0	1	2	3
1	0.00	19.51	36.93	50.71
2	0.00	19.48	36.79	50.39
3	0.00	19.43	36.50	49.71
4	0.00	19.32	36.00	48.60
5	0.00	19.16	35.21	46.95
6	0.00	18.87	33.99	44.60
7	0.00	18.39	32.11	41.09
8	0.00	17.52	29.19	36.00
9	0.00	15.88	24.50	24.50
10	0.00	12.50	12.50	12.50

(f) The optimal strategy can be derived from the table of $V(t, n)$ values in part 1(e). Notice that when $n = 1$, the expected profit increases as the concert date approaches, reaching its peak at $19.51 on day 1. For $n = 2$ or $n = 3$, the expected profit also increases as the concert date approaches, with the highest expected profit of $36.93 for $n = 2$ and $50.71 for $n = 3$ on day 1.

Therefore, if you have 1 ticket left, it is generally better to wait and sell the ticket closer to the concert date as the expected profit increases. If you have 2 or 3 tickets left, a similar strategy applies, with a preference for selling the tickets at a higher price as the concert date nears. This strategy takes advantage of the increasing probability of receiving a higher price as the concert date approaches while considering the available tickets to sell.