Tutorial 6: Dynamic programming II

1 A card game

Consider the card game where a sequence of n cards with values $v_1, \ldots, v_n$ are placed in a line on the table. Then two players take turns picking either the left-most or the right-most card left on the table.[1] At the end of the game, the player whose cards have the largest total value wins. In the event of a tie, both players are said to win. (In this game, we will always let n be even so that both players end up with the same number of cards.)

(i) The natural greedy strategy for Player 1 is to always take the card with the highest value (out of the left-most and the right-most cards available to be picked at the moment). Prove that this strategy does not guarantee that Player 1 always wins.

(ii) Design an algorithm with time complexity $\Theta(n^2)$ that computes a strategy for Player 1 that maximizes the total value of their cards, assuming that Player 2 always plays optimally. Given the initial sequence of card values $v_1, \ldots, v_n$, the algorithm should precompute some information in time $\Theta(n^2)$ in a way that afterwards, Player 1 can use this precomputed information to decide each move in time $\Theta(1)$.

(iii) Show that the strategy output by your algorithm guarantees that Player 1 will win the game.

1.1 Solution

(i) Consider the game with initial sequence of card values

$$2 \quad 10 \quad 1 \quad 1$$

on the table. The greedy strategy has Player 1 picking the card worth 2 points first, which causes the player to lose (3 points to 11). The alternative strategy, where Player 1 picks the fourth card instead causes Player 2 to be forced to "reveal" the card with value 10, no matter what choice is made, so that by then picking this card Player 1 wins 11 to 3.

(ii) For each $1 \leq i < j \leq n$, define $V[i, j]$ to be the difference in the total value of the cards picked by Player 1 and the total value of the cards picked by Player 2 when Player 1 plays an optimal strategy with cards of values $v_i, \ldots, v_j$ on the table, with $j - i + 1$ being an even number.

We can compute the values $V[i, j]$ in increasing order of $j - i$. The easy subproblems are when $j = i + 1$. In this case, there are only two cards left and Player 1s optimal strategy is to choose the one with the highest value so that

$$V[i, i+1] = \max\{v_i - v_{i+1}, v_{i+1} - v_i\} = |v_i - v_{i+1}|.$$

[1] I.e., if the cards $i, i+1, \ldots, j$ are left on the table, the player going next can take either card i or card j.

More generally, Player 1 has two choices: she can pick card v_i (at which point Player 2 can pick card v_{i+1} or card v_j) or card v_j (which lets Player 2 pick v_i or v_{j-1}). So

$$V[i,j] = \max \begin{cases} v_i + \min\{V[i+2,j] - v_{i+1}, V[i+1,j-1] - v_j\} \\ v_j + \min\{V[i+1,j-1] - v_i, V[i,j-2] - v_{j-1}\}. \end{cases}$$

We can also store the best choice to make at each step based on the values in V:

$$\text{NextCard}[i,i+1] = \begin{cases} i & \text{if } v_i \geq v_{i+1} \\ i+1 & \text{otherwise.} \end{cases}$$

and

$$\text{NextCard}[i,j] = \begin{cases} i & \text{if } v_i + \min\{V[i+2,j] - v_{i+1}, V[i+1,j-1] - v_j\}) \\ & \qquad \geq v_j + \min\{V[i+1,j-1] - v_i, V[i,j-2] - v_{j-1}\} \\ j & \text{otherwise.} \end{cases}$$

The algorithm for computing the arrays is as follows.

Algorithm 1: CardStrategy($v_1, \ldots, v_n$)

for $i = 1, \ldots, n-1$ **do**
    if $v_i \geq v_{i+1}$ **then**
        $V[i,i+1] \leftarrow v_i - v_{i+1}$;
        NextCard$[i,i+1] \leftarrow i$;
    else
        $V[i,i+1] \leftarrow v_{i+1} - v_i$;
        NextCard$[i,i+1] \leftarrow i+1$;
for $k = 3, 5, 7, 9, \ldots, n-1$ **do**
    for $i = 1, \ldots, n-k$ **do**
        $j \leftarrow i + k$;
        $w_1 \leftarrow v_i + \min\{V[i+2,j] - v_{i+1}, V[i+1,j-1] - v_j\}$;
        $w_2 \leftarrow v_j + \min\{V[i+1,j-1] - v_i, V[i,j-2] - v_{j-1}\}$;
        if $w_1 \geq w_2$ **then**
            $V[i,j] \leftarrow w_1$;
            NextCard$[i,j] \leftarrow i$;
        else
            $V[i,j] \leftarrow w_2$;
            NextCard$[i,j] \leftarrow j$;
return NextCard;

Theorem 1. *Assuming Player 2 plays optimally, the strategy in the NextCard array maximizes the total value of the cards picked up by Player 1.*

Proof. By induction on $j - i$, we can show that if only cards i through j remain on the table, following the NextCard strategy ensures Player 1 will earn as many points as possible (and the difference in points is stored in $V[i, j]$). If we prove this, the first claim is done, because we can substitute $i = 1$ and $j = n$.

The base case is when $j - i = 1$. In this case, NextCard$[i, j]$ is i if and only if $v_i \geq v_{i+1}$. Picking the card specified by NextCard$[i, j]$ will always pick the greater of the two choices.

In the inductive step, we assume that $j - i > 1$ and for every pair i', j' such that $j' - i' < j - i$, following the NextCard strategy ensures the maximum number of points for Player 1 on cards i' through j', and $V[i', j']$ stores the point difference.

The total value of the cards Player 1 obtains is exactly w_1 (as written in the CardStrategy algorithm) if NextCard$[i, j] = i$, and w_2 if NextCard$[i, j] = j$. In either case, the NextCard array is set to choose the greater of these two values. □

(iii) We have shown that the NextCard strategy maximizes the total value of cards picked up by Player 1. Second, we will show that there exists a strategy where Player 1 wins. Since the optimal strategy will earn at least as many points as the known winning strategy, the optimal strategy is winning as well.

Theorem 2. *There exists a strategy where Player 2 plays optimally but Player 1 still wins.*

Proof. Split the set of cards into the even-indexed cards and the odd-indexed cards, ie. $\mathcal{E} = \{v_i : i \text{ is even}\}$ and $\mathcal{O} = \{v_i : i \text{ is odd}\}$.

Let $S_e = \sum_{i \in \mathcal{E}} v_i$ and $S_o = \sum_{i \in \mathcal{O}} v_i$. Suppose that $S_e \geq S_o$. Then if Player 1 can pick up all of the even-indexed cards, Player 1 wins. Initially, n is an even number, so $j - i = n - 1$ is an odd number. Inductively, we have that every time it is Player 1's turn, $j - i$ is an odd number. Since $j - i$ is an odd number, exactly one of i and j is an even number. Therefore, Player 1 can pick one even-indexed card each turn.

Since Player 1 and Player 2 pick the same number of cards, and all of Player 1's cards are even-indexed, Player 1 has all of the even-indexed cards and Player 2 has all of the odd-indexed cards. Since $S_e \geq S_o$, Player 1 wins. The same argument applies in the case where $S_o \geq S_e$. □