Tutorial 1

Deepank Agrawal (17CS30011)

18-07-2019

1 Problem Statement

$A[1..m]$ and $B[1..n]$ are two 1D arrays containing m and n integers respectively, where $m \leq n$. We need to construct a sub-array $C[1..m]$ of B such that the **expression** $\sum_{i=1}^{m} \big|A[i] - C[i]\big|$ is minimized.

2 Recurrences

To solve the problem, we can design a Dynamic Programming(DP) algorithm. The formulation of the DP is:
Suppose that for any $i \,\epsilon\, [1, m]$ and $k \,\epsilon\, [1, n]$, $i \leq k$, we have already scanned arrays $A[1..i-1]$ and constructed $C[1..i-1]$, which is the optimal sub-array. Now, for $A[i]$, on scanning any $B[k] \,\forall\, k \,\epsilon\, [1, n]$, the following two cases arise for construction of the optimal sub-array C:

1. $B[k]$ is included in C.

2. $B[k]$ is not included in C.

If $B[k]$ is included in C, then $C[1..i-1] \cup B[k]$ should be the so-far optimal sub-array. Else, $B[k]$ is not included and $C[1..i-1]$ is the optimal sub-array. This is decided by whether or not including $B[k]$ minimizes $M[i][k]$ i.e., whether $\big|A[i] - B[k]\big| + M[i-1][k-1] < M[i][k-1]$ or not.

Let's consider an array $M[1..m][1..n]$, where $M[i][k]$ stores the minimal value of the expression for arrays $A[1..i]$ and $B[1..k]$. Now the recurrence relation for the DP, $\forall i \,\epsilon\, [0, m]$, $k \,\epsilon\, [0, n]$, can be defined as:

$$M[i][k] = \begin{cases} 0, & i = 0 \\ \infty, & i < k \\ min\left\{\big|A[i] - B[k]\big| + M[i-1][k-1],\, M[i][k-1]\right\}, & k \geq i \end{cases} \quad (1)$$

So, $M[m][n]$ will have the final minimum value of the expression.

3 Algorithm

Now that Optimal Substructure has been defined, let's design the algorithm. Before that, we can see there are **overlapping sub-problems** (e.g. $M[3][5]$ will be required for both $M[4][6]$ and $M[3][6]$). So, **memoization** will be used.

To solve the problem, we store the minimal value of the expression $\forall i,\, k,\, i \leq k$ using equation (1) and along-side store $B[k]$ to be included in the optimal sub-array. For this we construct a matrix $M[0..m][0..n]$, where $M[i][k]$ stores the minimal value of the expression for arrays $A[1..i]$ and $B[1..k]$ and an array $C[0..m]$ to store the included $B[k]$ values. Bottom-up approach will be used to fill matrix M.

The optimal sub-array C is updated when $M[i][k] \neq M[i][k-1]$ in a top-down fashion. For this, iterate matrix $M\,:\,(m,\,n) \rightarrow (1,\,1)$ and check for $M[i][k] \neq M[i][k-1]$ condition.

Pseudocode:

initialize $M[1..m][0..n]$ **with** ∞ **and** $M[0][0..n]$ **with** 0

for $i : 1 \rightarrow m$ **do** $\quad \{\mathit{Construct}\ M\}$

for $k : i \rightarrow n$ **do**

$M[i][k] \leftarrow min\left\{\left|A[i] - B[k]\right| + M[i-1][k-1], M[i][k-1]\right\}$

end for

end for

$i \leftarrow m\,,\, k \leftarrow n$

while $i \neq 0$ **do** $\quad \{\mathit{Construct}\ C\}$

if $M[i][k] \neq M[i][k-1]$**, then do**

$C[i] \leftarrow B[k]$

$i \leftarrow i - 1$

end if

$k \leftarrow k - 1$

end while

4 Demonstration

1. Let $A = [9, 10] \,\&\, B = [4, 9, 14]$
   Now, $M[1][1] = \min\{\left|A[1] - B[1]\right| + M[0][0],\, M[1][0]\}$
   or, $M[1][1] = \min\{\left|9 - 4\right| + 0,\, \infty\}$ or, $M[1][1] = \min\{5, \infty\} = 5$
   and, $C[1] = B[1] = 4$.
   Similarly, the matrix $M[0..m][0..n]$ will be:

   $$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \infty & 5 & 0 & 0 \\ \infty & \infty & 6 & 4 \end{bmatrix}$$

   Here, since $M[2][3] \neq M[2][2]$ so, $C[2] = B[3] = 14$
   and, $M[1][2] \neq M[1][1]$ so, $C[1] = B[2] = 9$
   Hence, $C = [9, 14]$ is the optimal sub-array.

2. Let $A = [2, 7, 2] \,\&\, B = [5, 3, 6, 8]$
   As above, constructing the matrix $M[0..m][0..n]$:

   $$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \infty & 3 & 1 & 1 & 1 \\ \infty & \infty & 7 & 2 & 2 \\ \infty & \infty & \infty & 11 & 8 \end{bmatrix}$$

   Here, $M[3][4] \neq M[3][3]$, $M[2][3] \neq M[2][2]$, $M[1][2] \neq M[1][1]$
   So, $C[3] = B[4] = 8$, $C[2] = B[3] = 6$, $C[1] = B[2] = 3$
   Hence, $C = [3, 6, 8]$ is the optimal sub-array.

3. Let $A = [9, 10, 12] \,\&\, B = [7, 6, 9, 8, 9, 12]$
   As above, constructing the matrix $M[0..m][0..n]$:

   $$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & 2 & 2 & 0 & 0 & 0 & 0 \\ \infty & \infty & 6 & 3 & 2 & 1 & 1 \\ \infty & \infty & \infty & 9 & 7 & 5 & 2 \end{bmatrix}$$

   Here, $M[3][6] \neq M[3][5]$, $M[2][5] \neq M[2][4]$, $M[1][3] \neq M[1][2]$
   So, $C[3] = B[6] = 12$, $C[2] = B[5] = 9$, $C[1] = B[3] = 9$
   Hence, $C = [9, 9, 12]$ is the optimal sub-array.

5 Time and space complexities

5.1 Time Complexity

Let's refer to the above mentioned pseudocode for calculation of time complexity. As we can see, from line no. 2 & 3, the for loops are of $O(m)\, and\, O(n)$ complexity respectively. Inside the nested for loops, the operation is of $O(1)$ time. Also, from line no. 8, the while loop is of $O(n)$ time. So, the total time complexity of the algorithm is $O(mn)$ because of the nested loops.

5.2 Space Complexity

As a matrix $M[0..m][0..n]$ is constructed, the space complexity of the algorithm will also be $O(mn)$.