FAKULTAS
ILMU
KOMPUTER

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

Convolution

DAA Term 2 2023/2024

Convolution for Generating Function

- Suppose we have two Generating Functions (GF) represent two sequences:
  - $P(x) = 1 + x + x^2 + x^3 + x^4 + \cdots$ for $P_n = \langle 1,1,1,1,1, \dots \rangle$
  - $Q(x) = 1 + 2x + 4x^2 + 8x^3 + 16x^4 + \cdots$ for $Q_n = \langle 1,2,4,8,16 \dots. \rangle$

- We can obtain a new sequence R_n from the convolution of P_n and Q_n such that $R_n = \langle 1 \cdot 1, (1 \cdot 2 + 1 \cdot 1), (1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1), \dots \rangle = \langle 1,3,7, \dots \rangle$.
- The corresponding GF is $R(x) = 1 + 3x + 7x^2 + 15x^3 + \cdots$

This is one example of the use of **convolution**, for multiplying polynomials, specifically that represent the sequence’s generating functions.

Convolution of Two Vectors

- The previous example following the definition of convolution of two vectors.
- Given two vectors $a = (a_0, a_1, a_2, \dots, a_{n-1})$ and $b = (b_0, b_1, b_2, \dots, b_{n-1})$ of length n, the convolution of a and b is a vector $c = a \times b$ such that:

$$c_i = \sum_{i=0}^{k} a_i b_{k-i}$$

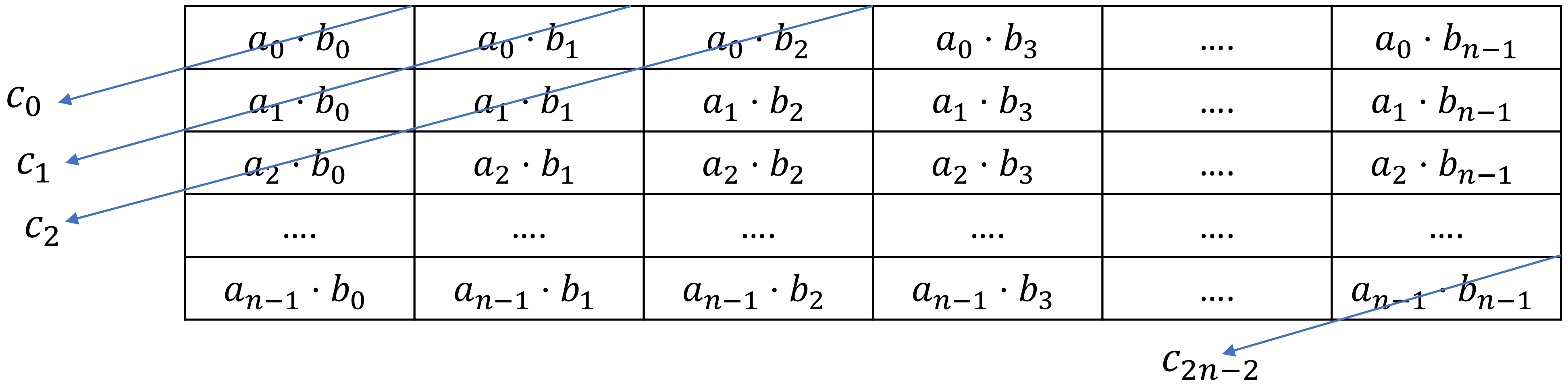
for $k = 0,1,2, \dots 2n - 2$

Then $c = (c_0, c_1, c_2, \dots, c_{2n-2})$

$$= \big((a_0 \cdot b_0), (a_0 \cdot b_1 + a_1 \cdot b_0), (a_0 \cdot b_2 + a_1 \cdot b_1 + a_2 \cdot b_0), \dots, (a_{n-1}, b_{n-1})\big)$$

Convolution of Two Vectors

- We can think $c = a \cdot b$ as an $n \times n$ table whose entry for cell (i, j) is $a_i \cdot b_j$ for $i, j = 0,1,2, \ldots n - 1$ and the convolution result is obtained by summing the diagonals as follows.



$a_0 \cdot b_0$	$a_0 \cdot b_1$	$a_0 \cdot b_2$	$a_0 \cdot b_3$	….	$a_0 \cdot b_{n-1}$
$a_1 \cdot b_0$	$a_1 \cdot b_1$	$a_1 \cdot b_2$	$a_1 \cdot b_3$	….	$a_1 \cdot b_{n-1}$
$a_2 \cdot b_0$	$a_2 \cdot b_1$	$a_2 \cdot b_2$	$a_2 \cdot b_3$	….	$a_2 \cdot b_{n-1}$
….	….	….	….	….	….
$a_{n-1} \cdot b_0$	$a_{n-1} \cdot b_1$	$a_{n-1} \cdot b_2$	$a_{n-1} \cdot b_3$	….	$a_{n-1} \cdot b_{n-1}$

Example

- Compute the convolution of $a = (1,2,2,3,4)$ and $b = (1,1,2,2,3)$

General Convolution of Two Vectors

- The more general case is the convolution of two vectors with different length.

- Given two vectors $a = (a_0, a_1, a_2, \dots, a_{m-1})$ and $b = (b_0, b_1, b_2, \dots, b_{n-1})$ of length n, the convolution of a and b is a vector $c = a \cdot b$ such that:

$$c_i = \sum_{i=0}^{k} a_i b_{k-i}$$

for $k = 0,1,2, \dots (m + n - 2)$

General Convolution of Two Vectors

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

- We can also solve a convolution of two vectors $a = (a_0, a_1, .., a_{m-1})$ and $b = (b_0, b_1, .., b_{n-1})$ by using the following steps:
  - Reverse the vector $b = (b_0, b_1, .., b_{n-1})$ to $b' = (b_{n-1}, b_{n-2}, .., b_1, b_0)$
  - Slide b' into successive positions relative to vector a for each successive value of the convolution, by summing the corresponding values of the two vectors.

$$\begin{array}{r} \\ b_{n-1}, b_{n-2}, .., b_1, \end{array} \boxed{\begin{array}{c} a_0 \\ b_0 \end{array}} \begin{array}{l} , a_1, .., a_{m-1} \\ \end{array} \longrightarrow c_0 = a_0 b_0$$

$$\begin{array}{r} \\ b_{n-1}, b_{n-2}, .., \end{array} \boxed{\begin{array}{c} a_0, a_1 \\ b_1, b_0 \end{array}} \begin{array}{l} , .., a_{m-1} \\ \end{array} \longrightarrow c_1 = a_0 b_1 + a_1 b_0$$

....

$$\begin{array}{r} a_0, a_1, a_2.., \\ \end{array} \boxed{\begin{array}{c} a_{m-1} \\ b_{n-1} \end{array}} \begin{array}{l} \\ , b_{n-2}, .. b_2, b_1, b_0 \end{array} \longrightarrow c_{m+n-2} = a_{m-1} b_{n-1}$$

Example

- Compute the convolution of $a = (2,2,3,3,4)$ and $b = (1,1,2)$

Application of Convolution

- Polynomial Multiplication
- Image/Signal Processing
- Combining Histogram
- String Matching
- Convolutional Neural Network

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Convolution for Image Smoothing

Original Image Filtered Image

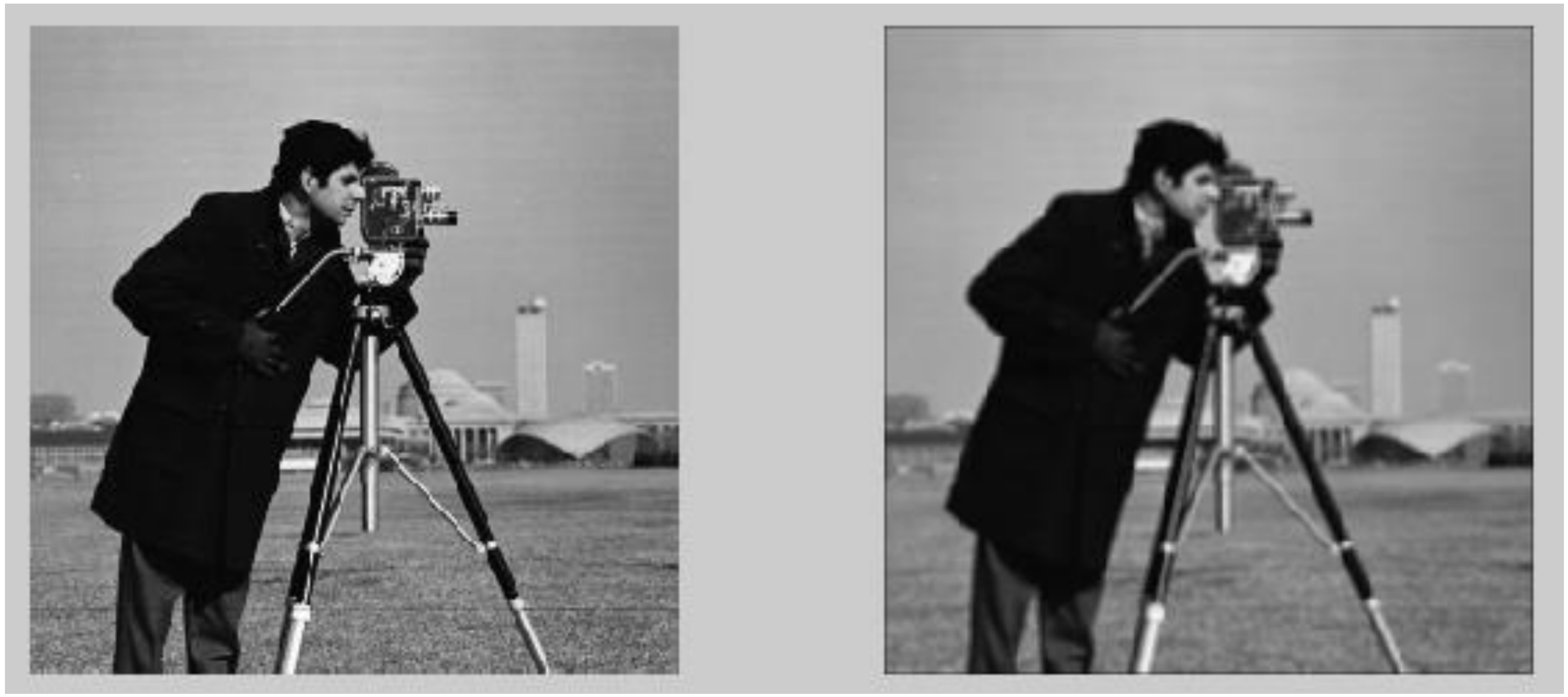


Image source: [Matlab Tutorial : Digital Image Processing 6 - Smoothing : Low pass filter - 2020 (bogotobogo.com)]

Convolution for Image Smoothing

- Suppose we have a vector represents a sequence of measurements, such as temperature or a stock price, sampled at m consecutive points in time. The vector is $a = (a_0, a_1, a_2, \dots, a_{m-1})$.

- This kind of sequences are often very noisy due to measurement errors or random fluctuations.
    - A common operation is needed to <u>smooth</u> the measurement by <u>averaging each value a_i with a weighted sum of its neighbors within k steps to left and right in the sequence</u>. The weights decaying quickly as one moves away from a_i.

Convolution for Image Smoothing

- Suppose a vector $w = \left(w_{-k}, w_{-(k-1)}, \ldots, w_{-1}, w_0, w_1, \ldots, w_{k-1}, w_k\right)$ represents a mask consisting of the weights we want to use for averaging each point with its neighbors.

We then iteratively position this mask so it is centered at each possible point in the sequence a; and for each positioning, we compute the weighted average. In other words, we replace a_i with

$$a_i' = \sum_{s=-k}^{k} w_s a_{i+s}.$$

Convolution for Image Smoothing

Let $b = (b_0, b_1,\ldots, b_{2k})$ by setting $b_t = w_{k-t}$. Then we have the smoothed value

$$a'_i = \sum_{(j,t):j+t=i+k} a_j b_t$$

So, the smoothed vector is just the convolution of the original vector and the reverse of the mask.

Convolution Implementation

```
# version 1
def convolution(a,b):
    na = len(a)
    nb = len(b)
    nc = na + nb - 1
    c = [0]*nc
    a = a + [0]*(nc-na)
    b = b + [0]*(nc-nb)

    for k in range(na+nb-1):
        for i in range(k+1):
            c[k] = c[k] + a[i]*b[k-i]
    return c
```

Convolution Implementation

```
# version 2
def convolution(a,b):
    na = len(a)
    nb = len(b)
    nc = na + nb - 1
    c = [0]*nc
    for k in range(nc):
        for i in range(k+1):
            if i < na and (k-i) < nb:
                c[k] = c[k] + a[i]*b[k-i]
    return c
```

References

- Lecturer Slides by Bapak L. Yohanes Stefanus for DAA Short Term 2022