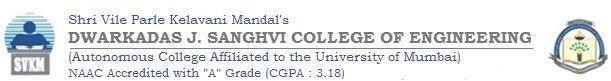
`

Department of Information Technology A.Y.

2024-2025

Class: TY BTech-IT, Semester: VI Subject: Big Data Lab

NAME: Anish Sharma SAP:60003220045

# Experiment – 10

1. Aim: To implement SON/CUR algorithm.

CODE:

SON:

import numpy as np

def son(A, num\_columns):

"""

Perform SON (Subsampled Orthogonalization and Normalization)

Args:

A: Input matrix (m x n)

num\_columns: Number of columns to select Returns:

A matrix C (m x num\_columns) representing selected columns """

# Step 1: Subsample the columns randomly m, n = A.shape

selected\_columns = np.random.choice(n, num\_columns, replace=False) C = A[:, selected\_columns]

# Step 2: Orthogonalization (using Gram-Schmidt or QR factorization) Q, R = np.linalg.qr(C)

# Step 3: Normalize the selected columns

C\_normalized = Q

return C\_normalized

# Example usage:

A = np.array([[1, 2, 3, 4],

[5, 6, 7, 8],

[9, 10, 11, 12]])

C\_son = son(A, num\_columns=2) print("SON Resulting Columns:\n", C\_son)

CUR:

def cur\_decomposition(A, num\_columns, num\_rows):

"""

Perform CUR Decomposition

Args:

A: Input matrix (m x n)

num\_columns: Number of columns to select num\_rows: Number of rows to select Returns:

C: Submatrix of selected columns

U: Middle matrix that approximates relationships

R: Submatrix of selected rows

""" m, n = A.shape

# Step 1: Select columns randomly (subsampling)

selected\_columns = np.random.choice(n, num\_columns, replace=False)

C = A[:, selected\_columns]

# Step 2: Select rows randomly (subsampling)

selected\_rows = np.random.choice(m, num\_rows, replace=False)

R = A[selected\_rows, :]

# Step 3: Calculate U using the pseudo-inverse of C and R

U = np.linalg.pinv(C) @ A @ np.linalg.pinv(R)

return C, U, R

# Example usage:

A = np.array([[1, 2, 3, 4],

[5, 6, 7, 8],

[9, 10, 11, 12]])

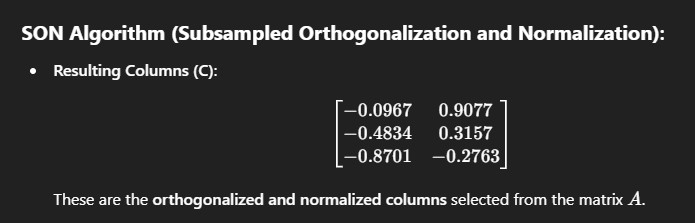
C\_cur, U\_cur, R\_cur = cur\_decomposition(A, num\_columns=2, num\_rows=2)

A\_approx = C\_cur @ U\_cur @ R\_cur

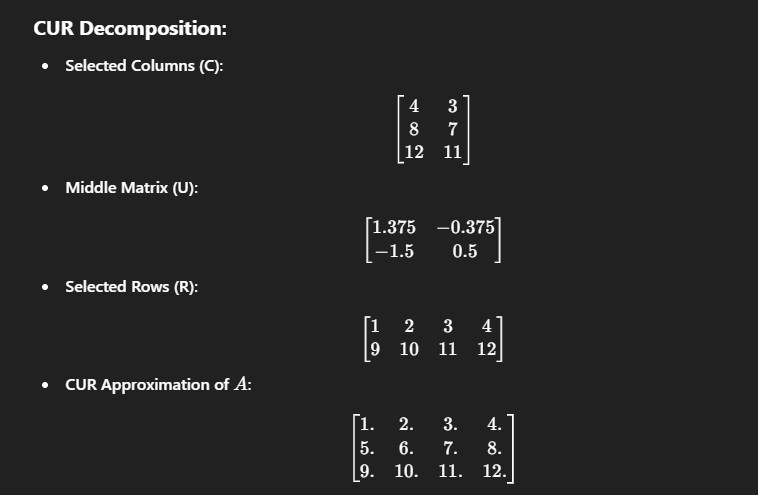
print("Original Matrix A:\n", A) print("CUR Decomposition - C:\n", C\_cur) print("CUR Decomposition - U:\n", U\_cur) print("CUR Decomposition - R:\n", R\_cur) print("CUR Approximation of A:\n", A\_approx)

1. Requirements: PC, Internet

OUTPUT: SON:



CUR:



1. Conclusion:

Thus, in this experiment, we implemented SON/CURE algorithms.