# **Department of Computer Science and Engineering**

**Data Science**

**Academic Year: 2024-2025 Name of Student:Diya Thakkar**

**Semester: 6 Student ID: 22107040**

**Class / Branch: CSE(DS) Date Of Performance: 10-3-25**

**Subject: ML Date Of Submission: 10-3-25**

**Name of Instructor: Prof.Ujwala Pagare**

**Experiment no. 6**

**Aim**:- To implement the Expectation-Maximization (EM) algorithm

**Program**:-

import numpy as np

# Given binary data

data = [

[1, 1, 1, 1, 0, 1, 1, 1, 1, 0],

[1, 1, 1, 1, 1, 1, 1, 0, 1, 1],

[1, 1, 1, 0, 1, 1, 1, 1, 0, 0],

[1, 1, 0, 1, 1, 1, 1, 0, 1, 1],

[1, 0, 1, 1, 0, 1, 1, 1, 1, 0]

]

# Initialize parameters

theta\_A, theta\_B = 0.6, 0.5 # Initial values

threshold = 1e-4 # Convergence threshold

max\_iterations = 100 # Maximum iterations allowed

# EM Algorithm

iteration = 0

while iteration < max\_iterations:

iteration += 1

prev\_theta\_A, prev\_theta\_B = theta\_A, theta\_B # Store previous values

# E-step: Compute responsibilities

P\_A, P\_B = [], []

for seq in data:

h = sum(seq)

t = len(seq) - h

# Compute likelihoods

L\_A = (theta\_A \*\* h) \* ((1 - theta\_A) \*\* t)

L\_B = (theta\_B \*\* h) \* ((1 - theta\_B) \*\* t)

# Normalize probabilities

denom = L\_A + L\_B

P\_A\_i = L\_A / denom if denom > 0 else 0.5

P\_B\_i = L\_B / denom if denom > 0 else 0.5

P\_A.append(P\_A\_i)

P\_B.append(P\_B\_i)

# M-step: Update theta\_A and theta\_B

total\_heads\_A = sum(P\_A[i] \* sum(data[i]) for i in range(len(data)))

total\_flips\_A = sum(P\_A[i] \* len(data[i]) for i in range(len(data)))

theta\_A = total\_heads\_A / total\_flips\_A if total\_flips\_A else prev\_theta\_A

total\_heads\_B = sum(P\_B[i] \* sum(data[i]) for i in range(len(data)))

total\_flips\_B = sum(P\_B[i] \* len(data[i]) for i in range(len(data)))

theta\_B = total\_heads\_B / total\_flips\_B if total\_flips\_B else prev\_theta\_B

# Check convergence

if abs(theta\_A - prev\_theta\_A) < threshold and abs(theta\_B - prev\_theta\_B) < threshold:

break

# Output final estimates

print(f"Stopped after {iteration} iterations.")

print(f"Estimated theta\_A: {theta\_A:.6f}")

print(f"Estimated theta\_B: {theta\_B:.6f}")

**Output**:-

