|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Hope Foundation’s**  **Finolex Academy of Management and Technology, Ratnagiri** | | | | | |
| **Department of Computer Science and Engineering (AIML)** | | | | | |
| Subject name: Machine Learning | | | | | | Subject Code: CSL604 | |
| Class | | TE CSE | | | Semester –VI (CBCGS) | Academic year: 2024-25 | |
| Name of Student | |  | | | | **QUIZ Score :6** | |
| Roll No | |  | | Experiment No. | | 05 | |
| Title: **To implement the Expectation MAximization algorithm.** | | | | | | | |
|  | | | | | | | |
| **1. Lab objectives applicable:**  **LOB1:**To introduce platforms such as Anaconda, COLAB suitable to Machine Learning.  **LOB4:** To implement Clustering techniques | | | | | | | |
| **2. Lab outcomes applicable:**  **LO2:** Apply suitable Machine learning models for a given problem.  **LO3:** Implement Neural Network based models. | | | | | | | |
| **3. Learning Objectives:**   1. To cluster the data points based on Gaussian Probability Distribution Function (PDF). | | | | | | | |
| **4. Practical applications of the assignment/experiment:**  To demonstrate Gaussian PDF based Expectation -Maximization algorithm | | | | | | | |
| **5. Prerequisites**:   1. Python language | | | | | | | |
| **6. Minimum Hardware Requirements**:-  I series processor, RAM 4GB,  **7. Software Requirements:-**  Colab or Visual Studio or Jupyter notebook (Anaconda) | | | | | | | |
| **8. Quiz Questions :** [**https://docs.google.com/forms/d/e/1FAIpQLSc\_S2hA2Cd38asYy-w3o6PTnpAoVvC-B0V\_Yl\_U\_ZqI6KFbQw/viewform?usp=dialog**](https://docs.google.com/forms/d/e/1FAIpQLSc_S2hA2Cd38asYy-w3o6PTnpAoVvC-B0V_Yl_U_ZqI6KFbQw/viewform?usp=dialog) | | | | | | | |
| **9. Experiment Evaluation:** | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | **Marks obtained** | **Out of** |
| **1** | Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used - | | | | | 6 | 6 |
| **2** | Lab Performance | | | | |  | 2 |
| **3** | Punctuality | | | | |  | 2 |
| **Date of performance (DOP)** | | |  | | **Total marks obtained** |  | **10** |

**Signature of Faculty**

**10. Theory:**

Expectation-Maximization is an iterative algorithm used to estimate parameters in statistical models particularly when there is hidden or missing data.

**Example:- Consider the following given data:[2.0,2.2,1.8,6.0,5.8.6.2]. Apply EM algorithm.**

1.Initiliase:

Mixing Coefficient:

Calculate

For

Similarly

For

Similarly

Step 2: Calculate Expectation using the formula:

Similarly,

Step 3: Maximization (Updation of Mean , Variance and Mixing Coefficient)

=2.001

Update Variance( Use new updated mean)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 0.391 | 0.000058 | 0.9998 | 0.000148 | 0 | 0 | -4 | 16 |
|  | 0.368 | 0.000133 | 0.9996 | 0.000361 | 0.2 | 0.04 | -3.8 | 14.44 |
|  | 0.398 | 0.000024 | 0.9999 | 0.000060 | -0.2 | 0.04 | -4.2 | 17.64 |
|  | 0.000058 | 0.391 | 0.000148 | 0.9998 | 4.0 | 16 | 0 | 0 |
|  | 0.000133 | 0.368 | 0.000361 | 0.9996 | 3.8 | 14.44 | -0.2 | 0.04 |
|  | 0.000024 | 0.398 | 0.000060 | 0.9999 | 4.2 | 17.64 | 0.2 | 0.04 |

Update Mixing Coefficient:

Perform these steps until convergence.

**11. Installation Steps / Performance Steps and Results –**

import numpy as np

from scipy.stats import norm

def expectation\_maximization(X, num\_clusters=2, max\_iters=100, tol=1e-4):

    np.random.seed(42)

    X = np.array(X)

    # Initialize means with the lowest and highest values in the dataset

    mu = np.array([np.min(X), np.max(X)])

    sigma = np.full(num\_clusters, 1.0)  # Initialize variances

    pi = np.full(num\_clusters, 1 / num\_clusters)  # Equal mixing coefficients

    print(f"Initial Means: {mu}, Initial Variances: {sigma}, Initial Mixing Coefficients: {pi}\n")

    for iteration in range(max\_iters):

        # Expectation Step

        responsibilities = np.zeros((len(X), num\_clusters))

        for j in range(num\_clusters):

            responsibilities[:, j] = pi[j] \* norm.pdf(X, mu[j], np.sqrt(sigma[j]))

        responsibilities\_sum = responsibilities.sum(axis=1, keepdims=True) + 1e-8

        responsibilities /= responsibilities\_sum  # Normalize

        # Maximization Step

        N\_k = responsibilities.sum(axis=0)

        N\_k = np.maximum(N\_k, 1e-8)  # Prevent zero division

        new\_mu = np.sum(responsibilities \* X[:, np.newaxis], axis=0) / N\_k

        new\_sigma = np.sum(responsibilities \* (X[:, np.newaxis] - new\_mu) \*\* 2, axis=0) / N\_k

        new\_pi = N\_k / len(X)

        # Print iteration-wise values

        print(f"Iteration {iteration + 1}:")

        print(f"Means: {new\_mu}")

        print(f"Variances: {new\_sigma}")

        print(f"Mixing Coefficients: {new\_pi}\n")

        # Convergence Check

        if np.allclose(mu, new\_mu, atol=tol) and np.allclose(sigma, new\_sigma, atol=tol):

            print("Converged!\n")

            break

        mu, sigma, pi = new\_mu, new\_sigma, new\_pi  # Update parameters

    return mu, sigma, pi, responsibilities

# Given dataset

X = [2.0, 2.2, 1.8, 6.0, 5.8, 6.2]

# Run EM algorithm

mu\_final, sigma\_final, pi\_final, responsibilities\_final = expectation\_maximization(X)

# Print final results

print(f"Final Means: {mu\_final}")

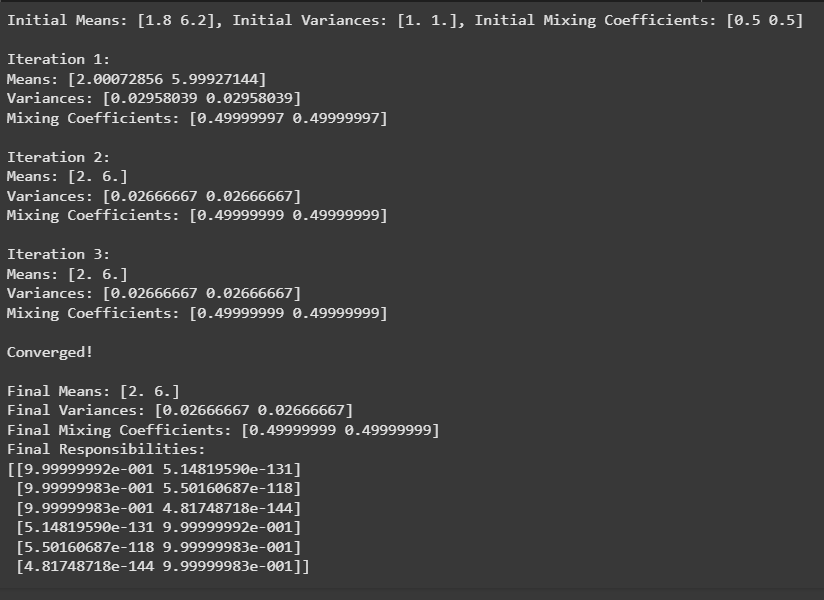
print(f"Final Variances: {sigma\_final}")

print(f"Final Mixing Coefficients: {pi\_final}")

print("Final Responsibilities:")

print(responsibilities\_final)

**Output:**

****

**12. Learning Outcomes Achieved**

1. Students are able to perform Expectation -MAximization algorithm based clustering.

**13. Conclusion:**

**1. Applications of the Studied Technique in Industry**

.**Data Clustering:** EM is used in **Gaussian Mixture Models (GMMs)** for customer segmentation, fraud detection, and anomaly detection.

**Natural Language Processing (NLP):** EM helps in **latent variable models**, such as Hidden Markov Models (HMMs), used in speech recognition and machine translation.

**Medical Imaging:** EM is applied in **image segmentation** and noise reduction in MRI and CT scan analysis.

**2. Engineering Relevance**

The EM algorithm is crucial in solving real-world engineering problems involving hidden variables and uncertainty in data. EM enhances unsupervised learning techniques, making it essential for AI applications.

**3. Skills Developed**

EM implementation helped in gaining deeper insights into probability theory, maximum likelihood estimation (MLE), and latent variable models and also improved programming knowledge.

**14. References**:

1. Nathalie Japkowicz & Mohak Shah, ―Evaluating Learning Algorithms: A Classification Perspective‖, Cambridge.
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖
3. Samir Roy and Chakraborty, ―Introduction to soft computing‖, Pearson Edition.
4. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher

Education

1. Peter Flach, ―Machine Learning‖, Cambridge University Press
2. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
3. Kevin P. Murphy, ―Machine Learning ― A Probabilistic Perspective‖, MIT Press