GMM

November 13, 2024

0.1 Generate the Synthetic Dataset

[3]:

```
import numpy as np

# Set random seed for reproducibility
np.random.seed(42)

# Parameters of the Gaussian mixture model
mu1, sigma1, pi1 = 0, 1, 0.3
mu2, sigma2, pi2 = 5, np.sqrt(2), 0.7
n_samples = 1000

# Generate samples
samples = []
for _ in range(n_samples):
    # Choose which component to sample from
    if np.random.rand() < pi1:
        samples.append(np.random.normal(mu1, sigma1))
    else:
        samples.append(np.random.normal(mu2, sigma2))

# Convert to a numpy array
samples = np.array(samples)
```

0.2 Implementing the EM Algorithm

[4]:

```
# Initial parameters
np.random.seed(42)
mu1_est, mu2_est = np.random.choice(samples, 2)  # random initial means
sigma1_est, sigma2_est = np.std(samples), np.std(samples)  # initial std␣
 ↪deviations
pi1_est, pi2_est = 0.5, 0.5  # initial mixing proportions

# EM algorithm parameters
max_iter = 100
tol = 1e-6
n = len(samples)
```

```python
# Helper function for Gaussian PDF
def gaussian_pdf(x, mean, std):
    return (1 / (np.sqrt(2 * np.pi) * std)) * np.exp(-0.5 * ((x - mean) / std) ** 2)

# EM algorithm
for iteration in range(max_iter):
    # E-step: Compute responsibilities
    r1 = pi1_est * gaussian_pdf(samples, mu1_est, sigma1_est)
    r2 = pi2_est * gaussian_pdf(samples, mu2_est, sigma2_est)
    total = r1 + r2
    gamma1 = r1 / total
    gamma2 = r2 / total

    # M-step: Update parameters
    N1 = np.sum(gamma1)
    N2 = np.sum(gamma2)

    # Update means
    mu1_new = np.sum(gamma1 * samples) / N1
    mu2_new = np.sum(gamma2 * samples) / N2

    # Update variances
    sigma1_new = np.sqrt(np.sum(gamma1 * (samples - mu1_new) ** 2) / N1)
    sigma2_new = np.sqrt(np.sum(gamma2 * (samples - mu2_new) ** 2) / N2)

    # Update mixing coefficients
    pi1_new = N1 / n
    pi2_new = N2 / n

    # Check for convergence
    if (
        np.abs(mu1_new - mu1_est) < tol and
        np.abs(mu2_new - mu2_est) < tol and
        np.abs(sigma1_new - sigma1_est) < tol and
        np.abs(sigma2_new - sigma2_est) < tol
    ):
        break

    # Update parameters for the next iteration
    mu1_est, mu2_est = mu1_new, mu2_new
    sigma1_est, sigma2_est = sigma1_new, sigma2_new
    pi1_est, pi2_est = pi1_new, pi2_new

# Final estimates
print(f"Estimated mu1: {mu1_est}, sigma1: {sigma1_est}, pi1: {pi1_est}")
print(f"Estimated mu2: {mu2_est}, sigma2: {sigma2_est}, pi2: {pi2_est}")
```

```
Estimated mu1: -0.019018690681499605, sigma1: 0.9416575998627877, pi1:
0.2926035819920777
Estimated mu2: 5.013590145860436, sigma2: 1.4295614962171463, pi2:
0.7073964180079222
```

0.3 Compare the estimated parameters with the true parameters and discuss the results.

0.4 Comparison of Estimated Parameters with True Parameters

0.4.1 True Parameters

- Mean of first Gaussian component (mu1): 0
- Standard deviation of first Gaussian component (sigma1): 1
- Mixing coefficient of first Gaussian component (pi1): 0.3
- Mean of second Gaussian component (mu2): 5
- Standard deviation of second Gaussian component (sigma2): sqrt(2) 1.414
- Mixing coefficient of second Gaussian component (pi2): 0.7

0.4.2 Estimated Parameters

- Mean of first Gaussian component (mu1_est): -0.019
- Standard deviation of first Gaussian component (sigma1_est): 0.942
- Mixing coefficient of first Gaussian component (pi1_est): 0.293
- Mean of second Gaussian component (mu2_est): 5.014
- Standard deviation of second Gaussian component (sigma2_est): 1.430
- Mixing coefficient of second Gaussian component (pi2_est): 0.707

0.4.3 Discussion

The estimated parameters are quite close to the true parameters, indicating that the EM algorithm has performed well in estimating the parameters of the Gaussian mixture model. Here are some observations:

- **Means (mu1 and mu2)**: The estimated means are very close to the true means. The slight deviation in `mu1_est` from `mu1` is minimal and can be attributed to the randomness in the data generation and the iterative nature of the EM algorithm.
- **Standard Deviations (sigma1 and sigma2)**: The estimated standard deviations are also close to the true values. The slight overestimation of `sigma2_est` compared to `sigma2` might be due to the variability in the data and the convergence criteria of the EM algorithm.
- **Mixing Coefficients (pi1 and pi2)**: The estimated mixing coefficients are very close to the true values, indicating that the algorithm has correctly identified the proportion of each component in the mixture.

Overall, the EM algorithm has successfully estimated the parameters of the Gaussian mixture model, with minor deviations that are expected in practical scenarios.