KHAI THÁC DỮ LIỆU

Nguyễn Xuân Việt Đức - 22280012

Bài tập thực hành - Lần 8

Bài 8: SỰ PHÂN TÍCH NHÓM (TIẾP THEO)

III. Nội dung thực hành:

1. Thuật toán Gaussian Mixture Model (GMM):

Gaussian Mixture Model (GMM) là một mô hình xác suất dùng để phân cụm (clustering) hoặc mô hình hóa phân phối dữ liệu. GMM giả định rằng dữ liệu được sinh ra từ sự kết hợp của nhiều phân phối chuẩn (Gaussian distributions), mỗi phân phối đại diện cho một cụm (cluster).

1.1. Khái niệm:

- GMM là sự kết hợp tuyến tính của nhiều phân phối chuẩn đa chiều.
- Mỗi thành phần (component) có các tham số: trọng số (weight), trung tâm (mean vector), và ma trận hiệp phương sai (covariance matrix).

1.2. Mô hình hóa xác suất:

- Xác suất của một điểm dữ liệu là tổng có trọng số của xác suất từ mỗi Gaussian component.

1.3. Ước lượng tham số:

- GMM sử dụng thuật toán EM (Expectation-Maximization) để tối ưu các tham số:
  - **E-step:** Tính xác suất mỗi điểm thuộc về từng component (soft assignment).
  - **M-step:** Cập nhật trọng số, mean, và covariance cho từng component dựa trên các xác suất này.

1.4. Ứng dụng:

- Phân cụm dữ liệu, phát hiện bất thường, trích xuất đặc trưng, mô hình hóa mật độ xác suất.

2. Thuật toán Mahalanobis k-means

2.1. K-means truyền thống

Mục tiêu: Phân cụm dữ liệu thành k nhóm sao cho tổng bình phương khoảng cách từ các điểm đến trung tâm cụm là nhỏ nhất.

Khoảng cách Euclid giữa điểm dữ liệu x và tâm cụm μ được tính như sau:

$$\boxed{d_E(x, \mu) = \sqrt{(x - \mu)^T (x - \mu)}}$$

2.2. Khoảng cách Mahalanobis:

Khoảng cách Mahalanobis giữa điểm x và tâm cụm μ được định nghĩa là:

$$\boxed{d_M(x, \mu) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}}$$

Trong đó:

- S là ma trận hiệp phương sai của cụm chứa μ
- S^{-1} là ma trận nghịch đảo của S

Khoảng cách Mahalanobis có khả năng:

- Xem xét mối tương quan giữa các thuộc tính
- Tự động chuẩn hóa dữ liệu theo phương sai

2.3. Mahalanobis K-means:

Ý tưởng: Thay khoảng cách Euclid trong thuật toán K-means bằng khoảng cách Mahalanobis.

Thuật toán:

1. Khởi tạo ngẫu nhiên k centroid.
2. Lặp lại cho đến khi hội tụ:
    - Gán mỗi điểm dữ liệu vào cụm có khoảng cách Mahalanobis nhỏ nhất.
    - Cập nhật centroid μ_j và tính lại ma trận hiệp phương sai S_j cho mỗi cụm.

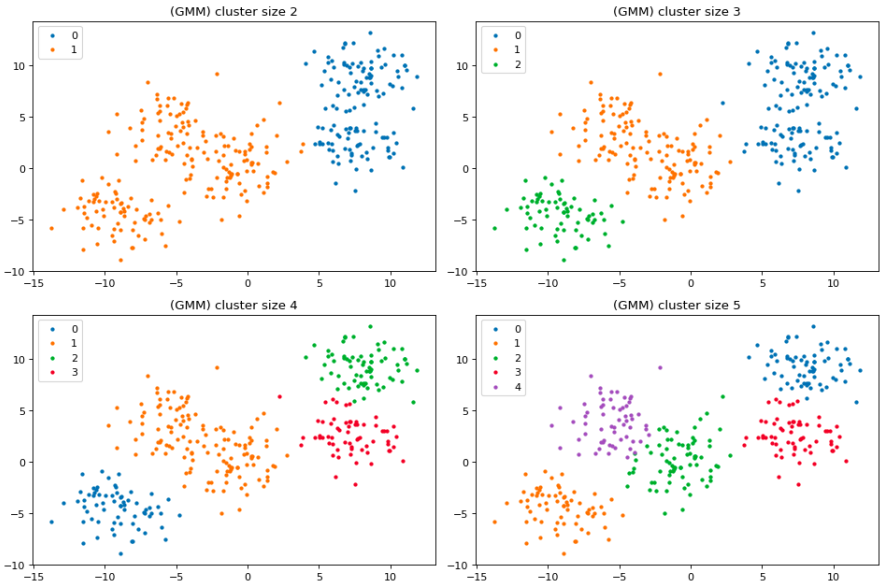
2.4. Nhược điểm

- Phức tạp tính toán do phải tính và nghịch đảo ma trận hiệp phương sai.
- Dễ gặp vấn đề nếu S không khả nghịch (đặc biệt khi số chiều lớn hơn số mẫu).

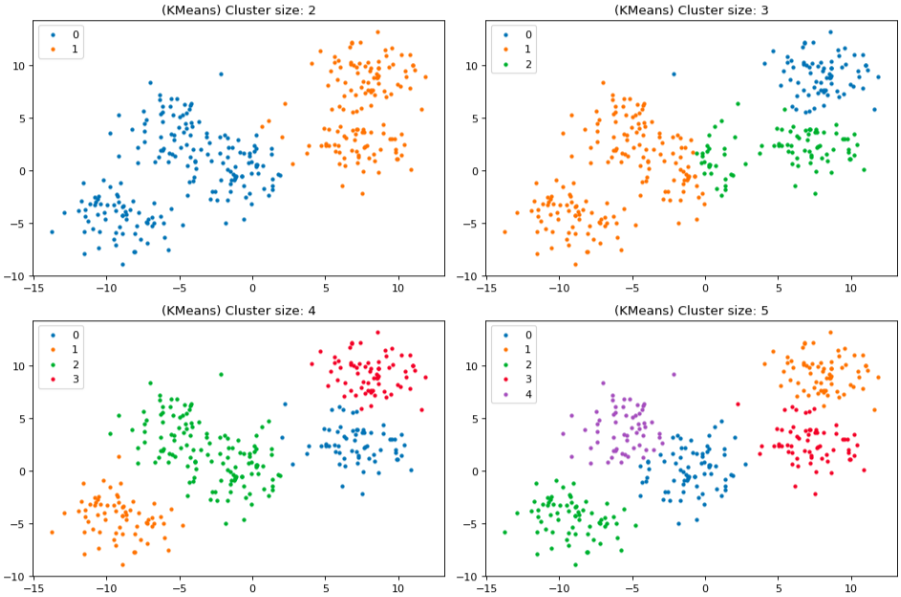
Implementation:

```python
def mahalanobis_kmeans(X, K, max_iters=100, tol=1e-4, random_state=None):
    n_samples, n_features = X.shape
    rng = np.random.default_rng(random_state)

    # 1. Pick K points at random as initial cluster centers
    indices = rng.choice(n_samples, K, replace=False)
    centers = X[indices]

    # 2. Calculate Euclidean distance to each cluster center and assign clusters
    dists = np.linalg.norm(X[:, np.newaxis, :] - centers[np.newaxis, :, :], axis=2)
    labels = np.argmin(dists, axis=1)

    for iteration in range(max_iters):
        prev_labels = labels.copy()

        # 3. Form clusters
        clusters = [X[labels == k] for k in range(K)]

        # 4. Calculate Mahalanobis distance and reassign clusters
        covariances = []
        for k in range(K):
            if len(clusters[k]) > 1:
                cov = np.cov(clusters[k], rowvar=False)
            else:
                # Regularize if too few points
                cov = np.eye(n_features)
            covariances.append(cov)

        mahal_dists = np.zeros((n_samples, K))
        for k in range(K):
            VI = np.linalg.pinv(covariances[k])
            for i in range(n_samples):
                mahal_dists[i, k] = mahalanobis(X[i], centers[k], VI)
        labels = np.argmin(mahal_dists, axis=1)

        # 5. Recalculate cluster means and covariances
        new_centers = np.array([X[labels == k].mean(axis=0) if np.any(labels == k) else centers[k] for k in range(K)])
        shift = np.linalg.norm(new_centers - centers)
        centers = new_centers

        if np.all(labels == prev_labels) or shift < tol:
            break

    return labels, centers, covariances
```

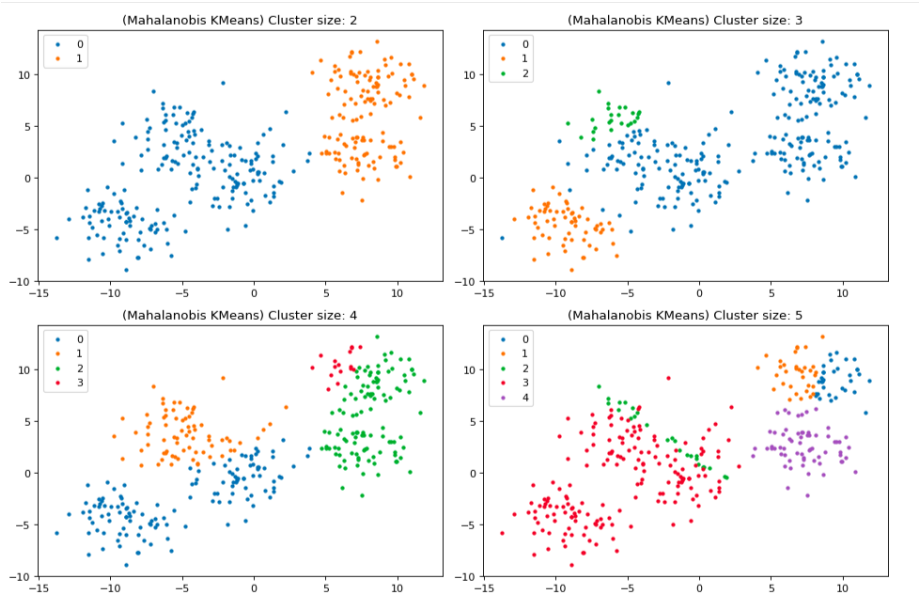
Kết quả: Gaussian Mixture Model



Kết quả: K-means



Kết quả: Mahalanobis K-means



Nhận xét tổng quan

K-means:

Ưu điểm:

- Đơn giản, dễ triển khai, tính toán nhanh.
- Hoạt động tốt khi cụm có dạng hình cầu và kích thước đồng đều.

Nhược điểm:

- Không mô hình hóa được các cụm có hình dạng phức tạp.
- Nhạy cảm với giá trị khởi tạo và nhiễu (outliers).
- Chỉ dùng khoảng cách Euclid, không xem xét mối tương quan giữa các thuộc tính.

Mahalanobis K-means

Ưu điểm:

- Tính đến mối tương quan giữa các thuộc tính qua ma trận hiệp phương sai.
- Hiệu quả hơn K-means khi các cụm có hình dạng không đồng nhất hoặc phương sai khác nhau.

Nhược điểm:

- Phức tạp hơn K-means, cần tính và nghịch đảo ma trận hiệp phương sai.
- Có thể gặp lỗi nếu dữ liệu có số chiều cao hơn số lượng mẫu.
- Không phải mô hình xác suất.

Gaussian Mixture Model (GMM)

Ưu điểm:

- Là mô hình xác suất, mô hình hóa cụm bằng các phân phối Gaussian.
- Cho phép gán mềm (soft assignment): mỗi điểm có xác suất thuộc về nhiều cụm.
- Linh hoạt, mô hình tốt cụm có hình dạng bất kỳ (tròn, oval, chéo...).

Nhược điểm:

- Tính toán phức tạp hơn (dùng thuật toán EM).
- Nhạy cảm với khởi tạo, dễ rơi vào cực trị địa phương (local optimum).
- Dễ bị overfitting nếu số cụm lớn hoặc dữ liệu ít.