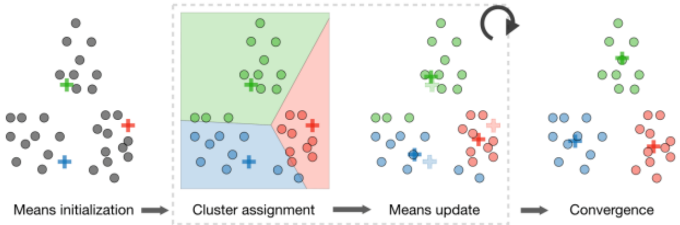
Unsupervised Learning

R Basheer Ahammad

The goal of unsupervised learning is to find hidden patterns in unlabeled data $\{x^{(1)}, \dots, x^{(m)}\}$.

1 K-means Clustering

```
Algorithm 1: K-means Clustering
Input  : Dataset $\{x^{(1)}, \dots, x^{(m)}\}$, Number of clusters $K$
Output : Cluster centroids $\{c_1, \dots, c_K\}$
Initialize cluster centroids $\{c_1^{(0)}, \dots, c_K^{(0)}\}$ randomly or using other methods;
while Not converged do
    for $i = 1$ to $m$ do
        Assign $x^{(i)}$ to the nearest centroid $c_k$ based on distance:
        $k^{(i)} = \arg\min_k \|x^{(i)} - c_k^{(t)}\|^2$;
    for $k = 1$ to $K$ do
        Update centroid $c_k^{(t+1)} = \frac{1}{\text{count}(c_k)} \sum_{i=1}^{m} x^{(i)}$ within cluster $k$;
```



Convergence conditions

Convergence in the k-means algorithm is achieved when the assignments of data points to clusters and the positions of cluster centroids remain unchanged or change within a small threshold between successive iterations.

Inter-Cluster Variance

The inter-cluster variance measures the spread between the cluster centroids and is defined as:

$$\text{Inter-Cluster Variance} = \sum_{i=1}^{K} n_i \cdot \|\mu_i - \mu\|_2^2$$

where K is the number of clusters, n_i is the number of data points in cluster i, μ_i is the centroid of cluster i, and μ is the overall mean of the data points.

Intra-Cluster Variance

The intra-cluster variance measures the spread within each cluster and is defined as:

$$\text{Intra-Cluster Variance} = \sum_{i=1}^{K} \sum_{x^{(j)} \in C_i} \|x^{(j)} - \mu_i\|_2^2$$

where C_i is cluster i, and μ_i is the centroid of cluster i.

2 Clustering Assessment Metrics

In an unsupervised learning setting, it is often hard to assess the performance of a model since we don't have the ground truth labels as was the case in the supervised learning setting.

Silhouette Score

$$\text{Silhouette Score}(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance of the data point i to other points in the same cluster, and $b(i)$ is the smallest average distance of the data point i to points in a different cluster.

Davies-Bouldin Index

$$\text{DBI} = \frac{1}{N} \sum_{i=1}^{N} \max_{j \neq i} \left(\frac{s_i + s_j}{d(i, j)} \right)$$

where s_i is the average distance of data point i to other points in the same cluster, and $d(i, j)$ is the distance between clusters i and j.

Calinski-Harabasz Index (Variance Ratio Criterion)

$$\text{CH Index} = \frac{B(k)}{W(k)} \times \frac{N - k}{k - 1}$$

where $B(k)$ is the between-cluster dispersion, $W(k)$ is the within-cluster dispersion, N is the number of data points, and k is the number of clusters.

3 Cluster Assignment Similarity Measures

Manhattan Distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Hamming distance

$$d(x, y) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

where,

$$\delta(x_i, y_i) = \begin{cases} 1, & \text{if } x_i \neq y_i \\ 0, & \text{if } x_i = y_i \end{cases}$$

Cosine Similarity

$$\text{similarity}(x,y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

4 Principal Component Analysis (PCA)

It is a dimension reduction technique that finds the variance maximizing directions onto which to project the data.

Eigen values, eigen vectors

Given a matrix $A \in \mathbb{R}^{n\times n}$, λ is said to be an eigenvalue of A if there exists a vector $\mathbf{z} \in \mathbb{R}^n \setminus \{0\}$, called an eigenvector, such that we have:

$$A\mathbf{z} = \lambda\mathbf{z}$$

Algorithm

The Principal Component Analysis (PCA) procedure is a dimension reduction technique that projects the data on k dimensions by maximizing the variance of the data as follows:

- Step 1: Normalize the Data Normalize the data to have a mean of 0 and standard deviation of 1:

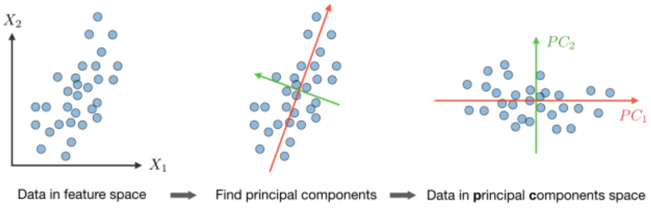
  $$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j}$$

  where $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$ and $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$.

- Step 2: Compute Covariance Matrix Compute $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i)T} \in \mathbb{R}^{n\times n}$, which is symmetric with real eigenvalues. which is symmetric with real eigenvalues.

- Step 3: Compute Principal Eigenvectors Compute $u_1, \ldots, u_k \in \mathbb{R}^n$, the k orthogonal principal eigenvectors of Σ, i.e., the orthogonal eigenvectors of the k largest eigenvalues.

- Step 4: Project Data onto Subspace Project the data on $\text{span}_{\mathbb{R}}(u_1, \ldots, u_k)$. This procedure maximizes the variance among all k-dimensional spaces.



5 Gaussian Mixture Model - GMM

Given a dataset $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, GMM aims to model the data as a mixture of K Gaussian distributions:

$$p(x) = \sum_{k=1}^{K} \phi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

where ϕ_k is the mixing coefficient for component k, and $\mathcal{N}(x|\mu_k, \Sigma_k)$ represents the Gaussian distribution with mean μ_k and covariance matrix Σ_k given by:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

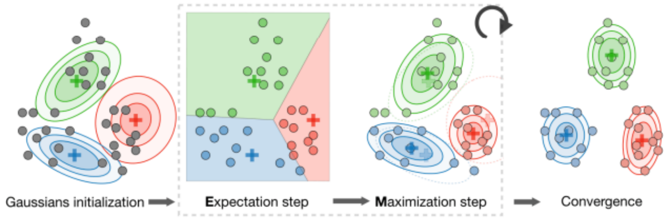
Where:

$\mathbf{x}$ is the vector of variables,
$\boldsymbol{\mu}$ is the mean vector,
$\boldsymbol{\Sigma}$ is the covariance matrix,
d is the dimension of the distribution,

The likelihood of the data given the model parameters can be expressed as:

$$p(X|\phi, \mu, \Sigma) = \prod_{i=1}^{m}\sum_{k=1}^{K} \phi_k \mathcal{N}(x^{(i)}|\mu_k, \Sigma_k)$$

The goal is to maximize this likelihood with respect to the parameters ϕ, μ, and Σ. The Expectation-Maximization (EM) algorithm is commonly used for GMM parameter estimation.



Expectation-Maximization (EM) Algorithm for GMM

The EM algorithm is commonly used to estimate the parameters of Gaussian Mixture Models. The algorithm alternates between the E-step and the M-step until convergence.

E-Step (Expectation): For each data point $x^{(i)}$, compute the posterior probabilities (responsibilities) for each component k:

$$\gamma_k^{(i)} = \frac{\phi_k \mathcal{N}(x^{(i)}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(x^{(i)}|\mu_j, \Sigma_j)}$$

M-Step (Maximization): Update the model parameters:

$$\phi_k = \frac{1}{m}\sum_{i=1}^{m}\gamma_k^{(i)}$$

$$\mu_k = \frac{\sum_{i=1}^{m}\gamma_k^{(i)}x^{(i)}}{\sum_{i=1}^{m}\gamma_k^{(i)}}$$

$$\Sigma_k = \frac{\sum_{i=1}^{m}\gamma_k^{(i)}(x^{(i)}-\mu_k)(x^{(i)}-\mu_k)^T}{\sum_{i=1}^{m}\gamma_k^{(i)}}$$

Repeat the E-step and M-step until the parameters converge.