Part 1: Introduction to Information Retrieval & Flat Clustering

Information Retrieval Overview

Information retrieval (IR) involves finding relevant documents from a large collection based on user queries. The core function of IR systems is to provide accurate and efficient access to data, often by ranking results based on relevance to the user's information need.

In unsupervised learning within IR, clustering is one of the most prominent techniques used to group similar documents. Unsupervised learning means that there is no predefined categorization or labels; instead, the algorithm uses the data's inherent properties to form clusters.

Clustering: Concepts and Definitions

Clustering involves dividing a set of documents into distinct groups (clusters) where documents within a cluster are more similar to each other than to those in other clusters. The two essential aspects of clustering are:

- **Intra-cluster similarity**: Documents within a cluster should be highly similar.
- **Inter-cluster dissimilarity**: Documents from different clusters should be distinct.

Types of Clustering

1. **Flat Clustering**: In flat clustering, the algorithm creates clusters without any hierarchy. Examples of flat clustering algorithms are:
    - **K-means** (hard clustering)
    - **Expectation-Maximization (EM)** (soft clustering)
2. **Hierarchical Clustering** (explored in Chapter 17 of the text): This method creates a tree-like structure of clusters where clusters can contain sub-clusters.

Clustering Hypothesis

The **Cluster Hypothesis** forms the foundation of using clustering in IR:

- **Cluster Hypothesis**: Documents in the same cluster behave similarly with respect to relevance to information needs.

This hypothesis posits that if one document from a cluster is relevant to a query, other documents from the same cluster are likely relevant as well. This is a critical assumption used to justify cluster-based retrieval systems.

Distance Measures in Clustering

The **distance measure** is the primary tool used to determine similarity between documents. Common distance measures include:

- **Euclidean distance**: Often used for clustering in document vector spaces.
- **Cosine similarity**: Frequently used when documents are represented as high-dimensional vectors, especially in text data.

Flat Clustering and K-means Algorithm

Flat clustering methods, such as K-means, do not impose a hierarchy on clusters. Instead, they group documents based on similarities and assign each document to exactly one cluster (in hard clustering). This differs from soft clustering, where a document can belong to multiple clusters with varying degrees of membership.

K-means Algorithm

The **K-means** algorithm is one of the most widely used flat clustering techniques because of its simplicity and efficiency. The algorithm works as follows:

1. **Initialization**: Randomly select K documents as the initial cluster centroids (seeds).
2. **Assignment**: Each document is assigned to the closest centroid based on the chosen distance metric.
3. **Recalculation**: The centroids of the clusters are recalculated based on the current assignments.
4. **Iteration**: Steps 2 and 3 are repeated until convergence, i.e., when document assignments no longer change.

Objective Function in K-means

The objective of K-means is to minimize the **residual sum of squares (RSS)**, which is defined as the sum of squared distances between each document and its assigned centroid. Mathematically:

$$RSS = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \omega_k} \|\mathbf{x} - \mu_k\|^2$$

Where:

- μ_k is the centroid of cluster ω_k,
- $\mathbf{x}$ is a document vector in the cluster.

By minimizing RSS, K-means ensures that the centroids represent the documents in their clusters as accurately as possible.

Example of K-means in Action (Numerical Illustration)

Let's consider a simple example of documents represented in a 2-dimensional vector space:

- Document 1: $(1, 2)$
- Document 2: $(2, 3)$
- Document 3: $(10, 10)$
- Document 4: $(11, 11)$

Assume we want to cluster these documents into 2 clusters. The initial cluster centroids are randomly selected as $(1, 2)$ and $(10, 10)$. The algorithm proceeds as follows:

1. **Initial Assignment**:
    - Document 1 is assigned to Cluster 1 (centroid $= (1, 2)$).
    - Document 2 is assigned to Cluster 1 (centroid $= (1, 2)$).
    - Document 3 is assigned to Cluster 2 (centroid $= (10, 10)$).
    - Document 4 is assigned to Cluster 2 (centroid $= (10, 10)$).
2. **Recalculate Centroids**:
    - New centroid of Cluster 1 = mean of Document 1 and Document 2 $= \left(\frac{1+2}{2}, \frac{2+3}{2}\right) = (1.5, 2.5)$.
    - New centroid of Cluster 2 = mean of Document 3 and Document 4 $= \left(\frac{10+11}{2}, \frac{10+11}{2}\right) = (10.5, 10.5)$.
3. **Repeat Assignment**:
    - Document assignments don't change as all documents are already assigned to the closest centroid.

Thus, the algorithm converges after one iteration, with the final clusters being:

- Cluster 1: {Document 1, Document 2}
- Cluster 2: {Document 3, Document 4}

Exercises

1. Define two documents as similar if they have at least two proper names in common (e.g., Clinton and Sarkozy). Give an example where the cluster hypothesis does not hold for this notion of similarity.

2. Create a simple one-dimensional example with two clusters where cluster-based retrieval is worse than direct nearest-neighbor search.

This concludes the first part of our deep dive into the theory and concepts of information retrieval clustering. In the next part, we will further explore advanced concepts like soft clustering, cluster cardinality, and model-based clustering.

Part 2: Soft Clustering, Cluster Evaluation, and K-means Extensions

Soft Clustering

Unlike **hard clustering**, where each document belongs to exactly one cluster, **soft clustering** allows for fractional memberships, meaning a document can belong to multiple clusters with varying degrees of membership. This is particularly useful when a document discusses multiple topics or themes. For instance, a document about the "Chinese automobile industry" could belong partially to both a "China" cluster and an "Automobile" cluster.

An important algorithm for soft clustering is the **Expectation-Maximization (EM)** algorithm, which generalizes K-means. Soft clustering allows for more nuanced clustering but can be more computationally intensive.

Expectation-Maximization (EM) Algorithm

The **Expectation-Maximization (EM)** algorithm is a soft clustering method that assumes the data is generated by a probabilistic model. The EM algorithm iteratively improves the parameters of the model by alternating between two steps:

1. **Expectation Step (E-step)**: This computes the probability that a document belongs to each cluster based on the current parameters.

2. **Maximization Step (M-step)**: This updates the parameters (centroids and cluster distributions) to maximize the likelihood of the data given the current assignments.

The EM algorithm can handle more complex data distributions than K-means because it models the data using probability distributions. It also allows fractional membership of documents in clusters, unlike K-means, which only assigns each document to one cluster.

Cluster Evaluation

Evaluating clustering quality is a challenging task, as we typically don't have ground truth labels for unsupervised learning tasks. However, several measures can be used to assess how well the clusters are formed.

Internal Evaluation Criteria

Internal criteria measure the quality of clusters based on the data itself. The two primary goals of a clustering algorithm are:

- **High intra-cluster similarity**: Documents within a cluster should be similar to each other.
- **Low inter-cluster similarity**: Documents from different clusters should be distinct.

K-means, for example, optimizes for **intra-cluster similarity** by minimizing the **residual sum of squares (RSS)** between the documents and their cluster centroids.

External Evaluation Criteria

External evaluation requires a ground truth classification (e.g., manually labeled clusters). The clustering output is compared to this reference classification. Four common external evaluation measures are:

1. **Purity**: This measure computes how well the clusters align with known categories by assigning each cluster to the most frequent class within it. It is calculated as:

$$\text{Purity} = \frac{1}{N} \sum_{k=1}^{K} \max_j |\omega_k \cap c_j|$$

where ω_k is a cluster and c_j is a class from the ground truth.

2. **Normalized Mutual Information (NMI)**: This information-theoretic measure assesses how much information is shared between the clustering and the true class labels. It is defined as:

$$\text{NMI}(W, C) = \frac{I(W; C)}{[H(W) + H(C)]/2}$$

where $I(W; C)$ is the mutual information between the clustering and the class labels, and $H(W)$ and $H(C)$ are the entropies of the clustering and the class labels.

3. **Rand Index (RI)**: The Rand Index measures the agreement between two partitions by comparing how pairs of documents are assigned to the

same or different clusters in the ground truth and the clustering result. It is computed as:

$$\text{RI} = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP and TN are the true positive and true negative counts, and FP and FN are false positives and false negatives, respectively.

4. **F-Measure**: The F-measure is a harmonic mean of precision and recall, which are adapted to evaluate clusters by comparing how documents are assigned to clusters and classes.

Numerical Illustration: Cluster Evaluation with Purity

Let's compute purity for a simple example. Suppose we have a set of 10 documents divided into two clusters, ω_1 and ω_2, and two true classes, c_1 and c_2, as follows:

- Cluster ω_1: Documents 1, 2, 3, 4, 5 (3 from class c_1, 2 from class c_2)
- Cluster ω_2: Documents 6, 7, 8, 9, 10 (4 from class c_2, 1 from class c_1)

To compute purity:

- For ω_1, the most frequent class is c_1 with 3 documents.
- For ω_2, the most frequent class is c_2 with 4 documents.

The purity is:

$$\text{Purity} = \frac{3 + 4}{10} = 0.7$$

Thus, the clustering has a purity of 0.7, indicating that 70% of the documents were correctly clustered according to the most frequent class in each cluster.

K-means: Convergence and Cardinality

Convergence in K-means

K-means always converges because the residual sum of squares (RSS) monotonically decreases with each iteration. However, it may converge to a local minimum, which depends on the initial centroids (or seeds) chosen at the start of the algorithm. If poor initial seeds are chosen, K-means may produce suboptimal clusters.

Time Complexity of K-means

The time complexity of K-means is $O(I \times K \times N \times M)$, where:

- I is the number of iterations,
- K is the number of clusters,

- N is the number of documents,
- M is the dimensionality of the feature space.

K-means is efficient and works well with large datasets, but the number of iterations I can increase if the clusters take longer to converge.

Cluster Cardinality and Choosing K

A major issue in clustering is determining the number of clusters K. While K-means requires K to be predefined, selecting K is often done heuristically or based on domain knowledge. There are several methods for estimating K:

- **Elbow Method**: By plotting RSS versus K, the "elbow" of the curve suggests a good choice for K.
- **Akaike Information Criterion (AIC)** and **Bayesian Information Criterion (BIC)**: These are information-theoretic measures that penalize model complexity.

Example: Determining Optimal K

Imagine you have a dataset of 1000 documents, and you compute RSS for different values of K. The results are:

K	RSS
2	2000
3	1500
4	1300
5	1250
6	1230

The **elbow method** suggests that $K = 4$ is a good choice because the rate of decrease in RSS slows down after 4 clusters.

In the next part, we will explore more advanced clustering techniques, including model-based clustering and hierarchical clustering, as well as further evaluation methods for clustering.

Part 3: Advanced Clustering Techniques and Model-Based Clustering

Model-Based Clustering

Model-based clustering is a probabilistic approach where we assume the data is generated by a mixture of underlying probabilistic models. The goal of the clustering algorithm is to infer the parameters of these models based on the observed data. In contrast to K-means, which assumes spherical clusters, model-based clustering can handle more complex shapes by modeling different distributions.

For example, if we assume the data follows a mixture of Gaussian distributions, the task is to estimate the parameters of these distributions and assign each document a probability of belonging to each cluster. This approach leads to **soft clustering**, where a document can belong to multiple clusters with different probabilities, unlike hard clustering in K-means where a document belongs to exactly one cluster.

Expectation-Maximization (EM) Algorithm for Model-Based Clustering

The **Expectation-Maximization (EM)** algorithm is the most common approach to model-based clustering. It iteratively alternates between estimating the probability that each document belongs to a given cluster (E-step) and updating the cluster parameters (M-step) to maximize the likelihood of the data.

Steps in the EM Algorithm

1. **Initialization**: Start with initial estimates for the parameters of the model (e.g., cluster means and variances).

2. **E-step (Expectation)**: Compute the probability (responsibility) that each document belongs to each cluster, given the current parameters of the model. For document d and cluster k, this probability is denoted by r_{nk}, and it is computed as:

$$r_{nk} = \frac{\alpha_k P(d_n|\theta_k)}{\sum_{j=1}^{K} \alpha_j P(d_n|\theta_j)}$$

   where α_k is the prior probability of cluster k, and $P(d_n|\theta_k)$ is the likelihood of the document given the parameters θ_k for cluster k.

3. **M-step (Maximization)**: Update the model parameters to maximize the likelihood of the observed data, based on the current responsibilities from the E-step. This involves updating the cluster means and covariances, as well as the cluster priors α_k.

4. **Repeat**: Iterate between the E-step and M-step until convergence, i.e., when the parameters no longer change significantly or the likelihood function converges.

Gaussian Mixture Model (GMM)

A common example of model-based clustering is the **Gaussian Mixture Model (GMM)**, where each cluster is assumed to follow a Gaussian distribution. GMM uses the EM algorithm to estimate the parameters (means, variances, and priors) of the Gaussians.

Example: EM for a Simple Gaussian Mixture

Suppose we have two clusters, each following a Gaussian distribution, and we want to use the EM algorithm to cluster the data.

1. **Initialization**: We start by guessing initial values for the means μ_1, μ_2, variances σ_1^2, σ_2^2, and priors α_1, α_2.

2. **E-step**: For each document, compute the probability of belonging to each cluster based on the current estimates of the Gaussian parameters. For a document d_n, the responsibility r_{nk} of cluster k is given by:

   $$r_{nk} = \frac{\alpha_k \mathcal{N}(d_n|\mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \alpha_j \mathcal{N}(d_n|\mu_j, \sigma_j^2)}$$

   Where $\mathcal{N}(d_n|\mu_k, \sigma_k^2)$ is the Gaussian probability density function.

3. **M-step**: Update the means, variances, and priors based on the current responsibilities:

   $$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} d_n}{\sum_{n=1}^{N} r_{nk}}$$

   $$\sigma_k^2 = \frac{\sum_{n=1}^{N} r_{nk}(d_n - \mu_k)^2}{\sum_{n=1}^{N} r_{nk}}$$

   $$\alpha_k = \frac{\sum_{n=1}^{N} r_{nk}}{N}$$

4. **Repeat**: Continue iterating between the E-step and M-step until the parameters converge.

Cluster Cardinality and Model Selection

An important question in clustering is how to choose the number of clusters, denoted by K. In K-means, we must specify K beforehand, but in model-based clustering, we can attempt to estimate the optimal number of clusters using model selection criteria such as:

1. **Akaike Information Criterion (AIC)**: This criterion balances goodness-of-fit (minimizing RSS) with model complexity (the number of clusters). AIC is given by:

   $$AIC = -2L + 2K$$

   where L is the log-likelihood of the model, and K is the number of parameters.

2. **Bayesian Information Criterion (BIC)**: Similar to AIC, but includes a stronger penalty for model complexity:

$$BIC = -2L + K \log N$$

where N is the number of documents. BIC tends to favor simpler models (i.e., fewer clusters) compared to AIC.

These criteria help avoid overfitting by penalizing models that use too many parameters, thus encouraging simpler models that generalize better to new data.

Exercises and Numerical Illustration

Exercise 1: EM for a Simple Gaussian Mixture

Given the following two clusters of 1-dimensional data:

- Cluster 1: $\mu_1 = 5$, $\sigma_1^2 = 1$
- Cluster 2: $\mu_2 = 10$, $\sigma_2^2 = 2$

1. Initialize the EM algorithm with random guesses for μ_1, μ_2, σ_1^2, σ_2^2, and α_1, α_2.
2. Run the E-step to compute the responsibilities for a set of documents, say $\{4, 6, 9, 11\}$.
3. Perform the M-step to update the cluster parameters based on the current responsibilities.

Exercise 2: Choosing K with BIC

Given a set of documents clustered with different values of K, compute the BIC for each K and select the optimal number of clusters. Use the formula:

$$BIC = -2L + K \log N$$

where L is the log-likelihood of the model for each K, and N is the number of documents.

Numerical Example for BIC Calculation

Suppose you have clustered a set of 100 documents with $K = 2, 3, 4, \ldots, 6$ clusters, and the log-likelihoods of the models are as follows:

K	Log-Likelihood (L)	BIC
2	-1200	?
3	-1100	?
4	-1050	?
5	-1040	?
6	-1035	?

To compute BIC for each K:

- For $K = 2$: $BIC = -2(-1200) + 2 \times \log 100 = 2400 + 9.21 = 2409.21$
- For $K = 3$: $BIC = -2(-1100) + 3 \times \log 100 = 2200 + 13.82 = 2213.82$
- For $K = 4$: $BIC = -2(-1050) + 4 \times \log 100 = 2100 + 18.42 = 2118.42$

Continue this calculation to find the optimal number of clusters (the model with the lowest BIC).

Hierarchical Clustering (Overview for Next Section)

While flat clustering (like K-means and EM) divides data into clusters without any relationships between them, **hierarchical clustering** builds a tree of clusters. Hierarchical clustering will be covered in the next part, where we will explore how to build dendrograms, interpret them, and apply hierarchical methods to different data sets.

In the next section, we will explore hierarchical clustering methods and discuss more sophisticated clustering evaluation techniques.

Part 4: Hierarchical Clustering and Advanced Evaluation Methods

Hierarchical Clustering

Unlike flat clustering (e.g., K-means, EM), which partitions data into distinct groups without any relationships between clusters, **hierarchical clustering** organizes data into a tree-like structure called a **dendrogram**. This structure shows how clusters are nested within each other at different levels of similarity. Hierarchical clustering is especially useful for exploring data when we want to visualize relationships between clusters or don't know the optimal number of clusters in advance.

There are two main types of hierarchical clustering:

1. **Agglomerative Hierarchical Clustering (Bottom-up)**: This starts with each document as its own cluster and iteratively merges the closest clusters until only one cluster remains.
2. **Divisive Hierarchical Clustering (Top-down)**: This starts with all documents in one cluster and iteratively splits them into smaller clusters until each document is in its own cluster.

Agglomerative Hierarchical Clustering (AHC)

Agglomerative clustering is the most common form of hierarchical clustering. The algorithm proceeds as follows:

1. Start with each document as a singleton cluster.
2. At each iteration, merge the two clusters that are most similar (according to a chosen similarity measure).
3. Repeat until all documents are grouped into a single cluster.

The result is a **dendrogram**, which is a tree that represents the merging process. By cutting the dendrogram at different levels, we can obtain different numbers of clusters.

Key Concepts in Agglomerative Clustering

- **Linkage Criteria**: The decision of which clusters to merge is based on a chosen linkage criterion:
    - **Single linkage**: The distance between two clusters is the minimum distance between any pair of documents, one from each cluster.
    - **Complete linkage**: The distance between two clusters is the maximum distance between any pair of documents, one from each cluster.
    - **Average linkage**: The distance between two clusters is the average of all pairwise distances between documents from both clusters.
- **Distance Metrics**: Hierarchical clustering can use different distance metrics:
    - **Euclidean distance**: Common for continuous data.
    - **Cosine similarity**: Often used for text data represented as vectors in high-dimensional space.

Example: Agglomerative Clustering with Single Linkage

Consider four documents represented in a 2D space:

- Document 1: (1, 2)
- Document 2: (2, 3)
- Document 3: (10, 10)
- Document 4: (11, 11)

Steps for agglomerative clustering using **single linkage**:

1. Initially, each document is its own cluster.
2. Compute the pairwise distances between all documents. The closest pair is Document 1 and Document 2.
3. Merge Document 1 and Document 2 into one cluster.

4. Repeat, merging the next closest pair (Document 3 and Document 4), until all documents are in one cluster.

The resulting **dendrogram** shows how clusters were merged over time, with the height of each merge representing the distance between the merged clusters.

Divisive Hierarchical Clustering

In **divisive clustering**, we start with a single cluster containing all documents and iteratively split the cluster into smaller sub-clusters. This process continues until each document is in its own cluster. Divisive clustering is less commonly used than agglomerative clustering due to its higher computational cost.

Cluster Evaluation in Hierarchical Clustering

Evaluating the quality of clusters in hierarchical clustering can be challenging due to the hierarchical structure. Two common approaches are:

1. **Internal criteria**: These measure the quality of clusters based on intra-cluster similarity and inter-cluster dissimilarity, similar to flat clustering.
2. **External criteria**: These compare the hierarchical clustering with a known ground truth (e.g., a classification of documents).

Cophenetic Correlation Coefficient

The **cophenetic correlation coefficient** is a measure of how faithfully a dendrogram represents the pairwise distances between documents. It compares the original pairwise distances with the cophenetic distances (i.e., the height at which each pair of documents is first grouped into the same cluster). A higher cophenetic correlation indicates that the dendrogram is a good representation of the data.

Cutting the Dendrogram

A critical step in using hierarchical clustering is deciding where to "cut" the dendrogram to obtain a desired number of clusters. The cut level can be chosen based on the **height** of the merge steps, which correspond to distances between clusters. Alternatively, external criteria like **silhouette scores** (which measure how well-separated clusters are) can help determine the optimal number of clusters.

Example: Cutting a Dendrogram

Imagine a dendrogram where documents are grouped as follows:

- Merge Documents 1 and 2 at height 1.
- Merge Documents 3 and 4 at height 1.5.

- Merge the clusters {1, 2} and {3, 4} at height 5.

By cutting the dendrogram at height 3, we obtain two clusters: {1, 2} and {3, 4}. Cutting at height 1 results in four singleton clusters, while cutting at height 5 results in a single cluster containing all documents.

Advanced Cluster Evaluation Methods

In addition to traditional evaluation metrics like purity and NMI, more sophisticated methods can be used to assess the quality of clustering.

Silhouette Score

The **silhouette score** measures how similar a document is to its own cluster compared to other clusters. For a document i, the silhouette score is computed as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where:

- $a(i)$ is the average distance between i and other documents in its cluster.
- $b(i)$ is the average distance between i and documents in the nearest neighboring cluster.

A silhouette score close to 1 indicates that the document is well-clustered, while a score close to -1 indicates that it is poorly clustered.

Dunn Index

The **Dunn Index** measures the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. A higher Dunn Index indicates better clustering:

$$\text{Dunn Index} = \frac{\min_{i \neq j} d(C_i, C_j)}{\max_k \Delta(C_k)}$$

where:

- $d(C_i, C_j)$ is the distance between clusters C_i and C_j.
- $\Delta(C_k)$ is the diameter of cluster C_k (i.e., the maximum distance between any two documents in the cluster).

Numerical Illustration: Silhouette Score Calculation

Suppose we have a set of documents clustered as follows:

- Cluster 1: {1, 2, 3}
- Cluster 2: {4, 5}

The pairwise distances between documents are:

	1	2	3	4	5
1	0	1	2	6	7
2	1	0	1.5	6.5	7.5
3	2	1.5	0	7	8
4	6	6.5	7	0	1
5	7	7.5	8	1	0

For Document 1:

- $a(1) = \frac{1+2}{2} = 1.5$ (average distance to other members of Cluster 1).
- $b(1) = 6$ (average distance to the nearest cluster, Cluster 2).

The silhouette score for Document 1 is:

$$S(1) = \frac{6 - 1.5}{\max(1.5, 6)} = \frac{4.5}{6} = 0.75$$

This indicates that Document 1 is well-clustered.

Hierarchical vs. Flat Clustering

Advantages of Hierarchical Clustering:

- **Flexibility**: No need to predefine the number of clusters.
- **Visual Insight**: Dendrograms provide a visual representation of cluster relationships.
- **Stability**: Hierarchical clustering is often more stable than flat clustering since it explores the data structure more comprehensively.

Disadvantages:

- **Computationally Intensive**: Hierarchical clustering has a higher computational cost compared to flat clustering, particularly for large datasets.
- **Irreversible Merges**: Once clusters are merged in agglomerative clustering, they cannot be split, which can lead to suboptimal clusters.

Conclusion

Hierarchical clustering offers a powerful tool for exploring data and uncovering its natural structure. By building a tree-like representation of the data, it provides valuable insights into relationships between documents and allows us to explore different clusterings at various levels of granularity. With appropriate evaluation methods like silhouette scores and Dunn Index, we can assess the quality of the resulting clusters and choose the best solution for a given task.

This completes the exploration of clustering techniques in information retrieval. We've covered everything from flat clustering and the K-means algorithm to more advanced techniques like model-based and hierarchical clustering. We've also explored a variety of cluster evaluation methods to ensure the best possible results.

Numerical Example 1: K-means Clustering (Moderate Difficulty)

You are given a set of 5 documents represented as 2-dimensional vectors in a feature space. Use the **K-means algorithm** to cluster these documents into 2 clusters.

The document vectors are:

- Document 1: (1, 2)
- Document 2: (3, 4)
- Document 3: (10, 10)
- Document 4: (12, 12)
- Document 5: (6, 8)

Step 1: Initialization

Initialize the centroids randomly. Let's assume we randomly select:

- Centroid 1: (1, 2) (same as Document 1)
- Centroid 2: (10, 10) (same as Document 3)

Step 2: Assignment Step

Compute the Euclidean distance between each document and both centroids.

$$\text{Distance from Document 1 to Centroid 1} = \sqrt{(1-1)^2 + (2-2)^2} = 0$$

$$\text{Distance from Document 1 to Centroid 2} = \sqrt{(1-10)^2 + (2-10)^2} = \sqrt{81+64} = \sqrt{145} \approx 12.04$$

Thus, Document 1 is assigned to Centroid 1.

Similarly, compute distances for the other documents:

- **Document 2**:

$$\text{To Centroid 1} : \sqrt{(3-1)^2 + (4-2)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

$$\text{To Centroid 2} : \sqrt{(3-10)^2 + (4-10)^2} = \sqrt{49+36} = \sqrt{85} \approx 9.22$$

Assigned to Centroid 1.

- **Document 3**:

$$\text{To Centroid 1}: \sqrt{(10-1)^2+(10-2)^2} = \sqrt{81+64} = \sqrt{145} \approx 12.04$$

$$\text{To Centroid 2}: \sqrt{(10-10)^2+(10-10)^2} = 0$$

  Assigned to Centroid 2.

- **Document 4**:

$$\text{To Centroid 1}: \sqrt{(12-1)^2+(12-2)^2} = \sqrt{121+100} = \sqrt{221} \approx 14.87$$

$$\text{To Centroid 2}: \sqrt{(12-10)^2+(12-10)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

  Assigned to Centroid 2.

- **Document 5**:

$$\text{To Centroid 1}: \sqrt{(6-1)^2+(8-2)^2} = \sqrt{25+36} = \sqrt{61} \approx 7.81$$

$$\text{To Centroid 2}: \sqrt{(6-10)^2+(8-10)^2} = \sqrt{16+4} = \sqrt{20} \approx 4.47$$

  Assigned to Centroid 2.

Step 3: Update Centroids

Update the centroids based on the mean of the documents in each cluster.

- For **Centroid 1**, the mean of the documents (Document 1, Document 2):

$$\mu_1 = \left(\frac{1+3}{2}, \frac{2+4}{2}\right) = (2, 3)$$

- For **Centroid 2**, the mean of the documents (Document 3, Document 4, Document 5):

$$\mu_2 = \left(\frac{10+12+6}{3}, \frac{10+12+8}{3}\right) = \left(\frac{28}{3}, \frac{30}{3}\right) = (9.33, 10)$$

Step 4: Repeat Assignment and Update

Using the new centroids, repeat the assignment and update steps until convergence.

Questions

1. Perform the next iteration of the assignment step using the updated centroids $(2, 3)$ and $(9.33, 10)$.
2. After reassignment, update the centroids again.
3. Determine if the algorithm has converged, or if further iterations are necessary.

This process should help consolidate your understanding of how K-means operates.

Numerical Example 2: EM Algorithm for Gaussian Mixture Model (Hard Difficulty)

You are given a dataset of 5 points in a 1-dimensional space. You will apply the **Expectation-Maximization (EM) algorithm** to cluster the points into 2 clusters, assuming that each cluster follows a Gaussian distribution.

The data points are:

- $x_1 = 2$
- $x_2 = 3$
- $x_3 = 7$
- $x_4 = 8$
- $x_5 = 10$

Step 1: Initialization

Let's start by randomly initializing the parameters for the two Gaussian distributions (clusters):

- Means $\mu_1 = 2$, $\mu_2 = 9$
- Variances $\sigma_1^2 = 1$, $\sigma_2^2 = 1$
- Priors (weights) $\alpha_1 = 0.5$, $\alpha_2 = 0.5$

Step 2: Expectation Step (E-Step)

In the E-step, we compute the responsibility of each cluster for each data point. This is the probability that a point belongs to a cluster, given the current parameters. The formula for the responsibility of cluster k for data point x_i is:

$$r_{ik} = \frac{\alpha_k \cdot \mathcal{N}(x_i|\mu_k, \sigma_k^2)}{\sum_{j=1}^{2} \alpha_j \cdot \mathcal{N}(x_i|\mu_j, \sigma_j^2)}$$

Where $\mathcal{N}(x_i|\mu_k, \sigma_k^2)$ is the Gaussian probability density function:

$$\mathcal{N}(x_i|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right)$$

For each data point, we compute the responsibilities for both clusters.

Example for $x_1 = 2$:

$$\mathcal{N}(2|\mu_1 = 2, \sigma_1^2 = 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(2-2)^2}{2(1)}\right) = \frac{1}{\sqrt{2\pi}} \approx 0.3989$$

$$\mathcal{N}(2|\mu_2 = 9, \sigma_2^2 = 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(2-9)^2}{2(1)}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{49}{2}\right) \approx 2.11 \times 10^{-8}$$

Now, compute the responsibilities:

$$r_{11} = \frac{0.5 \times 0.3989}{0.5 \times 0.3989 + 0.5 \times 2.11 \times 10^{-8}} \approx 1$$

$$r_{12} = \frac{0.5 \times 2.11 \times 10^{-8}}{0.5 \times 0.3989 + 0.5 \times 2.11 \times 10^{-8}} \approx 0$$

Repeat this for the other data points to compute all the responsibilities r_{ik}.

Step 3: Maximization Step (M-Step)

In the M-step, we update the parameters μ_1, μ_2, σ_1^2, σ_2^2, and α_1, α_2 using the current responsibilities.

The updated mean for each cluster k is computed as:

$$\mu_k = \frac{\sum_{i=1}^N r_{ik} x_i}{\sum_{i=1}^N r_{ik}}$$

The updated variance for each cluster is:

$$\sigma_k^2 = \frac{\sum_{i=1}^N r_{ik}(x_i - \mu_k)^2}{\sum_{i=1}^N r_{ik}}$$

The updated prior (weight) for each cluster is:

$$\alpha_k = \frac{\sum_{i=1}^N r_{ik}}{N}$$

Example: Update for Cluster 1

Let's compute the updated mean for Cluster 1 using the responsibilities calculated in the E-step:

$$\mu_1 = \frac{r_{11}x_1 + r_{21}x_2 + r_{31}x_3 + r_{41}x_4 + r_{51}x_5}{r_{11} + r_{21} + r_{31} + r_{41} + r_{51}}$$

Use the responsibilities calculated in the E-step to compute this, then update the variances and priors accordingly.

Step 4: Iterate

Repeat the E-step and M-step until the parameters converge (i.e., the changes in parameters are smaller than a predefined threshold).

Questions

1. Perform the E-step to compute the responsibilities for all data points after initialization.
2. Perform the M-step to update the means, variances, and priors.
3. Repeat this for a few iterations and observe the convergence of the algorithm.

This example demonstrates how the EM algorithm is used for Gaussian Mixture Models to find soft clusters in data.

Numerical Example 3: Hierarchical Clustering with Complete Linkage (Hard Difficulty)

You are given the following set of 6 documents, each represented as points in a 2-dimensional space. Your task is to apply **agglomerative hierarchical clustering** with **complete linkage** to form a dendrogram and group the documents.

The coordinates of the documents are as follows:

Document 1:	(1, 2)
Document 2:	(2, 3)
Document 3:	(4, 5)
Document 4:	(7, 8)
Document 5:	(8, 9)
Document 6:	(9, 10)

Step 1: Compute Pairwise Distances

First, compute the pairwise Euclidean distances between all documents. The distance between two documents (x_1, y_1) and (x_2, y_2) is given by:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Using this formula, we can calculate the distance between each pair of documents. For example:

$$\text{Distance between Document 1 and Document 2} = \sqrt{(2-1)^2 + (3-2)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$$

$$\text{Distance between Document 1 and Document 3} = \sqrt{(4-1)^2 + (5-2)^2} = \sqrt{9+9} = \sqrt{18} \approx 4.24$$

The complete pairwise distance matrix is shown below:

	D1	D2	D3	D4	D5	D6
D1	0	1.41	4.24	8.49	9.90	11.31
D2	1.41	0	2.83	7.07	8.49	9.90
D3	4.24	2.83	0	4.24	5.66	7.07
D4	8.49	7.07	4.24	0	1.41	2.83
D5	9.90	8.49	5.66	1.41	0	1.41
D6	11.31	9.90	7.07	2.83	1.41	0

Step 2: Start Clustering

Agglomerative hierarchical clustering starts with each document as its own cluster. The algorithm merges clusters iteratively, based on a linkage criterion. In **complete linkage**, the distance between two clusters is defined as the maximum distance between any pair of points, one from each cluster.

Iteration 1: Merge Closest Clusters

In the first iteration, find the pair of clusters (initially, individual documents) that are closest. In this case, the smallest distance is between Document 4 and Document 5, which is 1.41.

- Merge Document 4 and Document 5 into a single cluster: $\{4, 5\}$.

Iteration 2: Update Distance Matrix

After merging Documents 4 and 5, update the distance matrix by recalculating the distances between the new cluster $\{4, 5\}$ and the remaining documents. Since we are using **complete linkage**, the distance between the new cluster and any other document is the maximum distance between any member of the cluster and that document.

For example, the distance between the cluster $\{4, 5\}$ and Document 6 is:

$$\text{Distance} = \max(\text{Distance between D4 and D6}, \text{Distance between D5 and D6}) = \max(2.83, 1.41) = 2.83$$

Recalculate the distance matrix after this merge.

Iteration 3: Merge Next Closest Clusters

After updating the distance matrix, the next smallest distance is between Document 6 and the cluster $\{4, 5\}$, with a distance of 2.83.

- Merge Document 6 with $\{4, 5\}$ into a new cluster: $\{4, 5, 6\}$.

Iteration 4: Repeat Merging

Continue this process, merging the closest clusters at each iteration. The algorithm proceeds as follows:

- Merge Documents 1 and 2 (distance = 1.41).
- Merge the cluster $\{1, 2\}$ with Document 3 (distance = 2.83).
- Finally, merge the two large clusters: $\{1, 2, 3\}$ and $\{4, 5, 6\}$.

Step 3: Create the Dendrogram

The final step is to visualize the merging process using a **dendrogram**. A dendrogram is a tree diagram that shows the order in which clusters were merged and the distances at which they were merged. The height of each merge corresponds to the distance between the clusters at that step.

The structure of the dendrogram for this example would look like this:

1. $\{4, 5\}$ merged at height 1.41.
2. $\{4, 5, 6\}$ merged at height 2.83.
3. $\{1, 2\}$ merged at height 1.41.
4. $\{1, 2, 3\}$ merged at height 2.83.
5. Finally, $\{1, 2, 3\}$ and $\{4, 5, 6\}$ merged at height 7.07.

Questions

1. Perform all the steps to manually compute the distances and merges for this example.
2. Draw the final dendrogram that represents the merging process.
3. Compare how the clustering changes when using **single linkage** instead of **complete linkage**.

This example demonstrates how to apply agglomerative hierarchical clustering using complete linkage, how to update the distance matrix, and how to interpret the results through a dendrogram.

Numerical Example 4: Evaluating Clustering with Silhouette Score (Moderate Difficulty)

You are given a dataset of 7 points in a 2-dimensional space and the results of clustering those points into 2 clusters. Your task is to calculate the **Silhouette Score** for each point to evaluate the quality of the clustering. The silhouette score measures how well each point is clustered by comparing the cohesion within its own cluster to the separation from the nearest other cluster.

The coordinates of the points are:

- **Cluster 1**:
  - Point A: (1, 1)
  - Point B: (2, 2)
  - Point C: (3, 3)
- **Cluster 2**:
  - Point D: (8, 8)
  - Point E: (9, 9)
  - Point F: (10, 10)
  - Point G: (12, 12)

Step 1: Compute Intra-Cluster Distance (Cohesion)

For each point, calculate $a(i)$, which is the average distance between the point and all other points within the same cluster.

Example for Point A:

In Cluster 1, we need to compute the distances between Point A and Points B and C.

- Distance between A and B:

$$\text{Distance} = \sqrt{(2-1)^2 + (2-1)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$$

- Distance between A and C:

$$\text{Distance} = \sqrt{(3-1)^2 + (3-1)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

The average distance $a(A)$ is:

$$a(A) = \frac{1.41 + 2.83}{2} = \frac{4.24}{2} = 2.12$$

Repeat this for Points B and C to calculate $a(B)$ and $a(C)$.

Step 2: Compute Inter-Cluster Distance (Separation)

Next, calculate $b(i)$, which is the average distance between the point and all the points in the nearest other cluster (Cluster 2 in this case).

Example for Point A:

To calculate $b(A)$, compute the distances between Point A and all points in Cluster 2 (Points D, E, F, and G):

- Distance between A and D:

$$\text{Distance} = \sqrt{(8-1)^2 + (8-1)^2} = \sqrt{49+49} = \sqrt{98} \approx 9.90$$

- Distance between A and E:

$$\text{Distance} = \sqrt{(9-1)^2 + (9-1)^2} = \sqrt{64+64} = \sqrt{128} \approx 11.31$$

- Distance between A and F:

$$\text{Distance} = \sqrt{(10-1)^2 + (10-1)^2} = \sqrt{81+81} = \sqrt{162} \approx 12.73$$

- Distance between A and G:

$$\text{Distance} = \sqrt{(12-1)^2 + (12-1)^2} = \sqrt{121+121} = \sqrt{242} \approx 15.56$$

The average distance $b(A)$ is:

$$b(A) = \frac{9.90 + 11.31 + 12.73 + 15.56}{4} = \frac{49.50}{4} = 12.38$$

Repeat this for Points B and C to calculate $b(B)$ and $b(C)$.

Step 3: Compute Silhouette Score

The silhouette score for each point i is given by:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Example for Point A:

$$S(A) = \frac{12.38 - 2.12}{\max(2.12, 12.38)} = \frac{10.26}{12.38} \approx 0.83$$

Repeat this for Points B and C to calculate $S(B)$ and $S(C)$.

Step 4: Interpretation

- A silhouette score close to 1 indicates that the point is well-clustered.
- A score close to 0 means the point lies on the boundary between clusters.
- A negative score means the point has been misclassified and is closer to a different cluster than its assigned cluster.

Questions

1. Calculate the silhouette score for all points in Cluster 1 (A, B, C).
2. Determine whether Cluster 1 is well-separated from Cluster 2 based on the silhouette scores.
3. Compare the average silhouette score for Cluster 1 and Cluster 2.

This example shows how to compute and interpret silhouette scores to evaluate clustering quality. The silhouette score provides insight into both the cohesion within clusters and the separation between different clusters, offering a robust measure of clustering effectiveness.

Numerical Example 5: Expectation-Maximization (EM) for a Gaussian Mixture Model with Two Clusters (Hard Difficulty)

You are given a set of 6 data points in 1-dimensional space. These points are generated from a mixture of two Gaussian distributions. Your task is to apply the **Expectation-Maximization (EM) algorithm** to estimate the parameters of the Gaussian distributions (i.e., the means and variances), and to assign probabilities that each point belongs to each of the two clusters.

The data points are:

- $x_1 = 1$
- $x_2 = 2$
- $x_3 = 5$
- $x_4 = 6$
- $x_5 = 8$
- $x_6 = 9$

Step 1: Initialization

Initialize the parameters of the two Gaussian distributions randomly:

- Mean $\mu_1 = 2$, Mean $\mu_2 = 7$
- Variance $\sigma_1^2 = 1$, Variance $\sigma_2^2 = 1$
- Priors $\alpha_1 = 0.5$, $\alpha_2 = 0.5$

Step 2: Expectation Step (E-step)

In the E-step, compute the **responsibility** r_{ik}, which is the probability that data point x_i belongs to cluster k. The formula for the responsibility of cluster k for point x_i is:

$$r_{ik} = \frac{\alpha_k \cdot \mathcal{N}(x_i|\mu_k, \sigma_k^2)}{\sum_{j=1}^{2} \alpha_j \cdot \mathcal{N}(x_i|\mu_j, \sigma_j^2)}$$

Where $\mathcal{N}(x_i|\mu_k, \sigma_k^2)$ is the Gaussian probability density function:

$$\mathcal{N}(x_i|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right)$$

Example for $x_1 = 1$:

For Cluster 1:

$$\mathcal{N}(1|\mu_1 = 2, \sigma_1^2 = 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(1-2)^2}{2(1)}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\right) \approx 0.24197$$

For Cluster 2:

$$\mathcal{N}(1|\mu_2 = 7, \sigma_2^2 = 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(1-7)^2}{2(1)}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{36}{2}\right) \approx 1.53 \times 10^{-8}$$

Now, compute the responsibilities for x_1:

$$r_{11} = \frac{0.5 \times 0.24197}{0.5 \times 0.24197 + 0.5 \times 1.53 \times 10^{-8}} \approx 1$$

$$r_{12} = \frac{0.5 \times 1.53 \times 10^{-8}}{0.5 \times 0.24197 + 0.5 \times 1.53 \times 10^{-8}} \approx 0$$

Repeat this for the other data points x_2, x_3, x_4, x_5, and x_6 to compute the responsibilities r_{ik} for each point in both clusters.

Step 3: Maximization Step (M-step)

In the M-step, update the parameters (means, variances, and priors) based on the current responsibilities from the E-step.

- The new mean for cluster k is:

$$\mu_k = \frac{\sum_{i=1}^{N} r_{ik} x_i}{\sum_{i=1}^{N} r_{ik}}$$

- The new variance for cluster k is:

$$\sigma_k^2 = \frac{\sum_{i=1}^{N} r_{ik} (x_i - \mu_k)^2}{\sum_{i=1}^{N} r_{ik}}$$

- The new prior (weight) for cluster k is:

$$\alpha_k = \frac{\sum_{i=1}^{N} r_{ik}}{N}$$

Example for Cluster 1:

Using the responsibilities calculated in the E-step, update the mean for Cluster 1:

$$\mu_1 = \frac{r_{11} \cdot 1 + r_{21} \cdot 2 + r_{31} \cdot 5 + r_{41} \cdot 6 + r_{51} \cdot 8 + r_{61} \cdot 9}{r_{11} + r_{21} + r_{31} + r_{41} + r_{51} + r_{61}}$$

Repeat for the variance and the prior for Cluster 1, then do the same for Cluster 2.

Step 4: Iterate

Repeat the E-step and M-step until the parameters (means, variances, and priors) converge, i.e., when the changes in parameters become negligible.

Questions

1. Perform the E-step for all data points and calculate the responsibilities for both clusters.
2. Perform the M-step to update the parameters for both clusters.
3. Run another iteration of the EM algorithm and observe how the parameters evolve.
4. Determine if the algorithm has converged after a few iterations.

This example involves running the EM algorithm on a simple dataset and illustrates how to estimate the parameters of Gaussian mixture models using soft clustering techniques. The iterative process of the EM algorithm allows for the refinement of the cluster parameters based on probabilistic assignments of data points to clusters.