Unit 4
Clustering

- Evaluation of Clustering System

Objective

- Clustering Algorithms
- Evaluating the Clustering System

Evaluation of Clustering

- Evaluating the effectiveness of the clustering results, known as clustering evaluation or validation
- It can be used to determine which clustering algorithm is best suited for a particular dataset and task, and to tune the hyperparameters of these algorithms

Evaluation of Clustering

- Challenging due to
  - Since clustering is an unsupervised learning method, there are no ground truth labels against which the clustering results can be compared.
  - Determining the **correct** number of clusters or the **best** clustering is often a subjective decision, even for domain experts. What one considers as a meaningful cluster, another might dismiss as coincidental.

Evaluation of Clustering

- Challenging due to
  - In many real-world datasets, the boundaries between clusters are not clear-cut.
    - Some data points might sit at the boundary of two clusters and could be reasonably assigned to both.
  - Different applications might prioritize different aspects of clustering.
    - For example, in one application, it might be essential to have tight, well-separated clusters, while in another, capturing the overall data structure might be more important.

Types of Evaluation

- Two types of clustering evaluation measures (or metrics)

- **Internal measures** do not require any ground truth to assess the quality of clusters. They are based solely on the data and the clustering results.

- **External measures** compare the clustering results to ground truth labels.

Internal Evaluation Measures

- Most internal validation measures are based on the following two criteria:

- **Compactness measures** how closely related objects in the same cluster are.
  - Compactness can be measured in different ways, such as by using the variance of the points within each cluster, or computing the average pairwise distance between them.
- **Separation measures** how distinct or well-separated a cluster is from other clusters.
  - Examples for measures of separation include pairwise distances between cluster centers or pairwise minimum distances between objects in different clusters.

Silhouette Index

- The silhouette index (or score) measures the degree of separation between clusters by comparing each object's similarity to its own cluster against its similarity to objects in other clusters

Silhouette Coefficient

- We first define the silhouette coefficient of a data point $\mathbf{x_i}$ as:

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max\{a(\mathbf{x}_i), b(\mathbf{x}_i)\}}$$

- where:
  - $a(x_i)$ is the average distance between x_i and all the other data points in its cluster.

Silhouette Coefficient

- More formally, if point x_i belongs to cluster C_j, then

$$a(\mathbf{x}_i) = \frac{1}{|C_i| - 1} \sum_{\mathbf{x}_j \in C_i, j \neq i} d(\mathbf{x}_i, \mathbf{x}_j)$$

- where $d(x_i, x_j)$ is the distance between points x_i and x_j.
  - We can interpret $a(x_i)$ as a measure of how well point x_i is matched to its own cluster (the smaller the value, the better the match).
  - Note that $a(x_i)$ is not clearly defined for clusters with size 1, in which case we set $s(x_i) = 0$.

Silhouette Coefficient

- $b(x_i)$ is the average distance between x_i and the points in its neighboring cluster, i.e., the cluster whose points have the smallest average distance to x_i:

$$b(\mathbf{x}_i) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{\mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$$

- The silhouette coefficient ranges from **-1 to +1**, where a high value indicates that the point is well matched to its own cluster and poorly matched to neighboring clusters.

Silhouette index (SI)

- Based on the silhouette coefficients of the samples, we now define the silhouette index (**SI**) as the average of the coefficients over all the data points:

$$SI = \frac{1}{n} \sum_{i=1}^{n} s(\mathbf{x}_i)$$

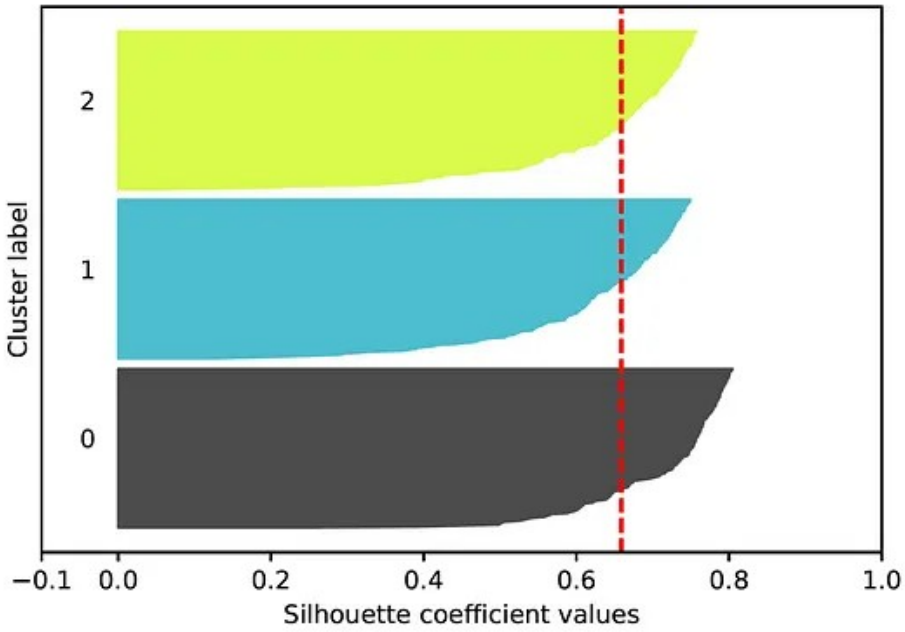
- where **n** is the total number of data points.

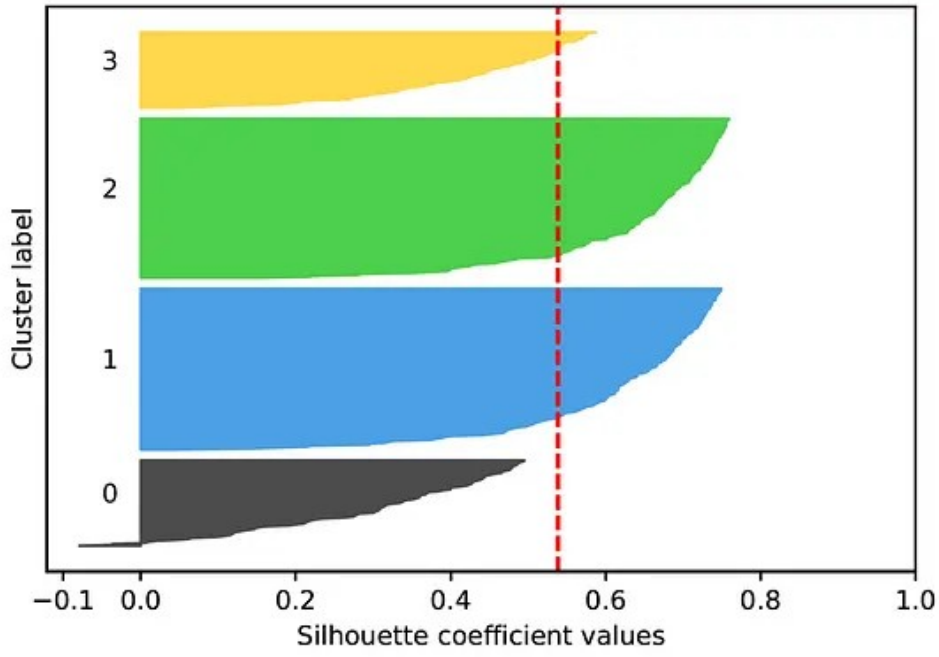
Silhouette index (SI)

- The silhouette index provides an overall measure for the quality of the clustering:
  - An index close to 1 means that the clusters are compact and well separated.
  - An index around 0 indicates overlapping clusters.
  - An index close to -1 means the clustering has either too many or too few clusters.

Evaluating based on SI

- K=3 vs. K=4





Implementation of SI

- Consider Separate Notebook for Demo

SI - Pros

- The score is easy to interpret.
  - It ranges from -1 to 1, where values close to 1 indicate well-separated clusters, and values close to -1 indicate poor clustering.
- It provides insight not only to the overall clustering quality, but also into quality of individual clusters.
  - This is often visualized using a Silhouette plot, which shows how each point in a cluster contributes to the overall score.
- It can be used to determine the optimal number of clusters in algorithms like k-means by comparing the scores for different values of k and taking the maximum score.
  - This approach tends to be more precise than the elbow method, which often requires a subjective judgement.

SI - Cons

- Tends to favor convex clusters and may not perform well with non-convex or irregularly shaped clusters.
- Does not consider the density of the clusters, which is important for evaluating density-based algorithms such as DBSCAN.
- When there is an overlap between the clusters, the silhouette score may provide ambiguous results.
- Can struggle with identifying subclusters within a larger cluster.
- Computationally intensive, as it requires computing all $O(n^2)$ pairwise distances between the points. This can make the evaluation process more costly than the clustering itself (when using k-means for example).
- Sensitive to noise and outliers, since it relies on the minimum pairwise distances, which can be affected by outliers.

Calinski-Harabasz Index

- Variance Ratio Criterion
- The Calinski-Harabasz index (CHI) measures the ratio between the between-cluster separation and the within-cluster dispersion:

$$CHI = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

- Where,
  - k is the number of clusters
  - n is the total number of data points
  - BCSS (Between-Cluster Sum of Squares)
  - WCSS (Within-Cluster Sum of Squares)

CHI

- BCSS (Between-Cluster Sum of Squares)
  - It is the weighted sum of squared Euclidean distances between each cluster centroid (mean) and the overall data centroid (mean):

$$BCSS = \sum_{i=1}^{k} n_i ||\mathbf{c}_i - \mathbf{c}||^2$$

  - where ni is the number of data points in cluster i,
  - ci is the centroid (mean) of cluster i, and c is the overall centroid (mean) of all data points.
- BCSS measures how well the clusters are separated from each other (the higher the better).

CHI

- WCSS (Within-Cluster Sum of Squares) is the sum of squared Euclidean distances between the data points and their respective cluster centroid:

$$WCSS = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} ||\mathbf{x} - \mathbf{c}_i||^2$$

- WCSS measures the compactness or cohesiveness of the clusters (the smaller the better).
- Minimizing WCSS (also known as the inertia) is the objective of centroid-based clustering such as k-means.

Explanation of CHI

- The numerator of CHI represents the between-cluster separation normalized by its degrees of freedom k - 1 (fixing the centroids of k - 1 clusters also determines the kth centroid, since its value makes the weighted sum of all centroids match the overall data centroid).

- On the other hand, the denominator of CHI represents the within-cluster dispersion normalized by its degrees of freedom n - k (fixing the centroid for each cluster reduces the degrees of freedom by one for each cluster).

$$CHI = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

Explanation of CHI

- Dividing both the BCSS and WCSS by their degrees of freedom helps to normalize the values, making them comparable across different numbers of clusters.
- Without this normalization, the CH index could be artificially inflated for higher values of k, making it hard to determine whether an increase in the index value is due to genuinely better clustering or just due to the increased number of clusters.
- A higher value of CHI indicates a better clustering, because it means that the data points are more spread out between clusters than they are within clusters.

CHI - Pros

- Simple to calculate and computationally efficient.
- Easy to interpret. Higher values generally indicate better clustering.
- Like silhouette score, it can be used to find the optimal number of clusters.
- Has some theoretical justification (it is analogous to the F-test statistic in univariate analysis, which compares variance between groups to variance within groups).

CHI - Cons

- Tends to favor convex clusters, and may not perform well with clusters irregular shapes.
- May not perform well with clusters of varying sizes as the variance within large clusters can disproportionately affect the balance between the BCSS and the WCSS.
- Not well-suited for evaluating density-based clustering like DBSCAN.
- Sensitive to noise and outliers, as these can significantly affect both the within-cluster and between-cluster dispersions.

Implementation of CHI

- Consider Separate Notebook for Demo

Davies-Bouldin Index

- The Davies-Bouldin index (DBI) measures the average similarity between each cluster and its most similar cluster,
  - where the similarity is defined as the ratio between
    - the intra-cluster distances (distances between the points in the cluster to the cluster center) and
    - the inter-cluster distance (distance between the cluster centers).
    -
    -
  - Where,
    - S_i is the average distance of all points in cluster i to the centroid of that cluster c_i
    - $d(c_i, c_j)$ is the distance between the centroids of clusters i and j.

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{d(\mathbf{c}_i, \mathbf{c}_j)}\right)$$

BDI – S_i - Spread of the cluster

- S_i measures the **spread** or **size** of the cluster.
- S_i is the average distance of all points in cluster i to the centroid of that cluster c_i

$$S_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{c}_i)$$

Explanation of DBI

- The ratio (Si + Sj) / d(ci, cj) measures the similarity between clusters i and j in terms of how close or overlapping the clusters are

- The numerator, which is the sum of the spreads of the two clusters, is high when both clusters are **large** (i.e., have large internal distances)

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{d(\mathbf{c}_i, \mathbf{c}_j)}\right)$$

Explanation of DBI

- The denominator, which is the distance between the cluster centroids, is low when the clusters are close to each other.

- Therefore, if the ratio between the numerator and the denominator is large, the two clusters may be overlapping or not well-separated. Conversely, if the ratio is small, it suggests that the clusters are well-separated relative to their size.

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{d(\mathbf{c}_i, \mathbf{c}_j)}\right)$$

Explanation of DBI

- For each cluster i, the index identifies the cluster j that maximizes this ratio, i.e., the cluster most similar to cluster i. The final DB index is the average of these worst-case similarities across all clusters.

- Thus, lower values of DB index indicate a better clustering with clusters that are compact and well-separated, where 0 is the lowest possible value.

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{d(\mathbf{c}_i, \mathbf{c}_j)}\right)$$

DBI - Pros

- Easy and fast to compute.
- Easy to interpret. Lower values of the index indicate better clustering, and a value of 0 indicates the ideal clustering.
- Like the previous two scores, it can be used to find the optimal number of clusters.

DBI - Cons

- Tends to favor convex clusters, and may not perform well with clusters of varying sizes or irregular shapes.
- Less effective for evaluating density-based clustering algorithms like DBSCAN.
- Noise and outliers can significantly impact the index.

Implementation of DBI

- Consider Separate Notebook for Demo

External Evaluation Measures

- External evaluation measures are used when the true labels of the data points are known.
- These measures compare the the results of the clustering algorithm against the ground truth labels.

Contingency Matrix

- Similar to confusion matrices in classification problems, a contingency matrix (or table) describes the relationship between the ground truth labels and the cluster labels.
- The rows of the matrix represent the ground-truth classes and its column represent the clusters.
- Each cell in the matrix, denoted by n_{ij}, contains the count of the data points that have a class label i and were assigned to cluster j.
- Various external evaluation metrics, such as adjusted Rand index and Fowlkes-Mallows index use the contingency matrix as the basis for their calculation.

Rand Index (ARI)

- The Rand Index (RI), named after William Rand, measures the similarity between the cluster assignments and the true class labels by making pairwise comparisons.

- It is calculated as the ratio of the number of agreements between the cluster assignments and the class labels to the total number of pairs of data points:

$$RI = \frac{a+b}{\binom{n}{2}}$$

Rand Index (ARI)

$$RI = \frac{a+b}{\binom{n}{2}}$$

- Where,
  - **a** is the number of pairs of points that have the same class label and also belong to the same cluster.
  - **b** is the number of pairs of points that have different class labels and belong to different clusters.
  - **n** is the total number of points.
- The RI ranges from 0 to 1, where 1 indicates that the cluster assignments and the class labels are exactly the same.

RI - problem

- In this case, a = 2 and b = 8, thus RI = 0.667

$$RI = \frac{2+8}{\binom{6}{2}} = \frac{10}{15} = 0.667$$

- The problem with Rand Index is that it can yield high values even for random cluster assignments, particularly when the number of clusters is large.
  - This is because when the number of clusters increases, the probability of randomly assigning points with different labels to different clusters increases.
  - Consequently, a specific RI value can be ambiguous, as it is not clear how much of the score is due to chance versus actual agreement.

Adjusted Rand index (ARI)

- The adjusted Rand index (ARI) corrects for this by normalizing the RI score, taking into account the expected RI score if the cluster assignments were made randomly. It is computed as follows:

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]}$$

- Where,
  - E[RI] is the expected value of the Rand index under random cluster assignment.
  - This value is calculated using the contingency table described above.
    - More specifically, we first calculate the sums of each row and each column in the table

ARI

- The ARI values range from -1 to 1, where 1 indicates perfect agreement between the cluster assignments and the class labels, 0 indicates random agreement, and negative values indicate agreement less than expected by chance.

ARI - Pros

- RI scores are bounded between 0 and 1, and ARI scores are bounded between -1 and 1.
  - The bounded range makes it easy to compare the scores between different algorithms.
- Random (uniform) cluster assignments have an ARI score close to 0 for any number of samples and clusters.
- The adjustment of ARI for chance makes it more reliable and interpretable.
- No assumption is made on the cluster structure, which makes these metrics useful for comparing different clustering algorithms independent of the cluster shapes.

ARI - Cons

- Do not take into account how data points are distributed within each cluster.

- For example, even if the data points within a cluster are spread out or form sub-clusters, this will not affect the RI or ARI scores.

Fowlkes-Mallows Index (FMI)

- The Fowlkes-Mallows Index (FMI) is defined as the geometric mean of the pairwise precision (the accuracy of grouped pairs of points) and recall (the completeness of correctly grouping pairs that belong together):

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}}$$

- where:
  - TP (True Positive) is the number of pairs of points that have the same class label and belong to the same cluster.
  - FP (False Positive) is the number of pairs of points that have different class labels but are assigned to the same cluster.
  - FN (False Negative) is the number of pairs of points that have the same class labels but are assigned to different clusters.

FMI

- The FMI score ranges from 0 to 1, where 0 indicates no correlation between the clustering results and the true labels, and 1 represents a perfect correlation.

FMI - Pros

- Considers both precision and recall, offering a balanced view of clustering performance.
- The scores are bounded between 0 and 1.
- Random (uniform) label assignments have a FMI score close to 0 for any number of samples and clusters.
- Does not make assumptions on the cluster structure.

FMI - Cons

- It is based on the analysis of pairs of elements, which may not capture the broader structural properties of the clusters, such as their shape or distribution.

- When the dataset is highly imbalanced (i.e., one class dominates the dataset), FMI might not accurately reflect the effectiveness of the clustering.

Homogeneity

- Homogeneity measures whether each cluster contains only members of a single class. It is defined as follows:

$$h = 1 - \frac{H(C|K)}{H(C)}$$

- where:
  - C represents the true class labels.
  - K represents the cluster labels assigned by the algorithm.
  - H(C|K) is the weighted average of the conditional entropies of the class distributions given the cluster assignments:

Homogeneity

$$H(C|K) = \sum_{k=1}^{|K|} P(K=k) H(C|K=k) = -\sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{n_{c,k}}{n} \log\left(\frac{n_{c,k}}{n_k}\right)$$

- Where $n_{c,k}$ is the number of samples from class c assigned to cluster k, n_k is the number of samples in cluster k and n is the total number of samples.

- H(C) is the entropy of the class distribution:

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} \log\left(\frac{n_c}{n}\right)$$

Homogeneity

- The homogeneity score ranges from 0 to 1, where 1 indicates perfect homogeneity, i.e., each cluster contains only members of a single class.

Completeness

- Completeness measures whether all members of a given class are assigned to the same cluster. It is defined as follows:

$$c = 1 - \frac{H(K|C)}{H(K)}$$

- Where
  - H(K|C) is weighted average of the conditional entropies of the cluster distributions given the class labels
  - H(K) is the entropy of the cluster distribution

Completeness

- Like homogeneity, completeness ranges from 0 to 1, where 1 indicates perfect completeness, i.e., each class members are assigned to a single cluster.

V-measure

- V-measure is the harmonic mean of homogeneity and completeness, providing a single score to assess the clustering performance:

$$v = 2 \cdot \frac{hc}{h + c}$$

- By using the harmonic mean, the V-measure penalizes imbalances between homogeneity and completeness, encouraging a more even clustering performance.