Clustering Exercise

Joan Caballero Castro

Julia Amenós Dien

Marc Gonzalez Vidal

Oriol Miró López-Feliu

November 2024

Introduction to Machine Learning (IML)

Practical Work 3: W3

MAI - UPC / UB / URV

Contents

1 Introduction

In this study, we implemented and analyzed various clustering algorithms using several datasets from the UCI repository. We assessed the behavior of the algorithms and the quality of their clustering using multiple clustering validation metrics. Additionally, we explored how the choice of metric influences the resulting clustering. Our analysis revealed that internal metrics produce different clustering outcomes compared to external metrics. For each algorithm, we analyzed how the clustering varied depending on the selected metric and how well the results aligned with the true labels. This comprehensive evaluation highlights the impact that metric selection has on clustering performance and its correspondence with actual class labels.

2 Datasets

We selected three datasets for this study:

- **Satimage:** A large numerical dataset containing 6,435 satellite image instances across 6 classes.
- **Splice:** A large categorical dataset comprising 3,190 DNA sequence instances spanning 3 classes.
- **Vowel:** A small mixed dataset with 990 instances of British English vowel sounds in 11 classes.

The preprocessing was applied uniformly across all datasets. Numerical features were standardized using StandardScaler, removing the mean and scaling to unit variance. Categorical features were one-hot encoded to convert them into a suitable numerical format for clustering algorithms. Duplicate rows were removed to eliminate redundant information and prevent skewing the clustering results.

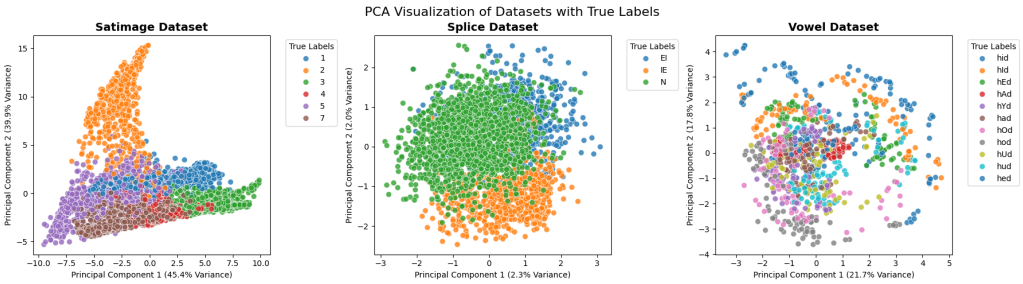

Figure 1: PCA visualization of clusters for Satimage, Splice, and Vowel datasets using true labels. Variance percentages for each Principal Component are in brackets.

Figure 1 visualizes the clusters within a PCA-reduced feature space based on true class labels. The **Satimage** dataset presents cohesive clusters, although there is significant overlap in the center and bottom regions. **Splice** shows considerable overlap in the central area, while **Vowel** presents many small and sparse clusters with high overlap. This high overlap in the Splice and Vowel datasets poses challenges for algorithms to generate effective clusters.

The variance explained by the first two principal components is crucial for interpretation. Satimage retains most of its structure with 45.36% and 39.88% of the variance explained by the first and second principal components, respectively, allowing for clear visualization of clusters. In contrast, Splice explains only 2.30% and 2.03% of the variance, and Vowel explains 21.74% and 17.76%, respectively. The limited variance captured by the first two components in Splice and Vowel significantly restricts cluster separation in two dimensions.

3 Experimental Approach

To evaluate performance, we utilize four widely-used clustering validation metrics: Adjusted Rand Index (ARI), Silhouette Coefficient, Davies-Bouldin Index (DBI), and Purity; each metric offers a unique perspective on the quality of the clustering.

3.1 Evaluation Metrics

Adjusted Rand Index (ARI) The ARI measures the similarity between the true labels and the predicted clustering, corrected for chance. Introduced by Hubert and Arabie [1], it provides a value in the range $[-1, 1]$, where values closer to 1 indicate a high degree of agreement. An ARI of 0 suggests random clustering, while negative values indicate disagreement.

Silhouette Coefficient The Silhouette Coefficient assesses both the cohesion within clusters and the separation between clusters. Proposed by Rousseeuw [2], it computes a value in the range $[-1, 1]$, where higher values imply better-defined clusters. A Silhouette score close to 1 indicates well-separated and compacted clusters, while values near -1 suggest overlapped clusters.

Davies-Bouldin Index (DBI) The DBI evaluates the average similarity ratio of each cluster with its most similar cluster, where similarity is defined as the ratio of intra-cluster to inter-cluster distances. The metric was introduced by Davies and Bouldin [3]. Lower DBI values are preferred, indicating more compact and well-separated clusters.

Purity The Purity assesses the extent to which each cluster contains only members of a single class, providing a measure of clustering quality. Purity computes the proportion of correctly assigned data points by assigning to each cluster the class most frequent within it, ranging from 0 to 1. Values close to 1 indicate that clusters are predominantly composed of a single class, while values near 0 suggest that clusters contain a mix of multiple classes.

3.2 Experimental Setup

To account for the stochastic nature of algorithms such as K-Means, we conduct the experiments using 5 different random seeds, specifically seed $\in \{0, 1, 2, 3, 4\}$. For each configuration under evaluation, we consider the results from the seed that produces the best outcomes.

To identify the optimal parameter configuration, we select the configuration that achieves the highest performance based on the chosen evaluation metric. For instance, when using the ARI, higher values indicate better clustering, so the configuration with the highest ARI score is selected. Conversely, for the DBI, lower values are preferable, and the configuration with the lowest DBI score is chosen. For algorithms that require parameter tuning, we explore a range of parameter values and systematically document the configurations that yield the best performance according to the selected metrics.

Each dataset is preprocessed as accorded to Section 2. We present the clustering results visually where possible and quantitatively using the aforementioned metrics to draw comparisons across datasets and algorithms.

4 OPTICS

In this section, we explore the application of the Ordering Points To Identify the Clustering Structure (OPTICS) clustering algorithm proposed by Ankerst et al. [4].

4.1 Experiments and Results

We evaluated the performance of the OPTICS algorithm on the Satimage, Splice, and Vowel datasets. The parameters explored were $metric \in \{'euclidean', 'manhattan', 'chebyshev', 'l1'\}$, $algorithm \in \{'ball\_tree', 'kd\_tree', 'brute'\}$, and $min\_samples\_list \in [2, 15]$. We explored additional parameters and in greater detail than those specified in the assignment to obtain more comprehensive results.

Metric determines how distances are calculated, and we selected a variety of metrics to evaluate the impact of different distance computation approaches. *Algorithm* specifies the method for finding nearest neighbors; we chose all three available options ($'ball\_tree'$, $'kd\_tree'$, $'brute'$) since the fourth option, $'auto'$, automatically selects one of these based on the dataset. *Min_samples* defines the minimum number of samples required in a neighborhood for a point to be considered a core point; we explored a wide range of values due to the varying sizes of our datasets.

Dataset	**Metric**	*metric*	*algorithm*	*min_samples*	**N. Clusters**	**Metric Score**
Satimage	ARI	euclidean	brute*	2	648	0.02208
	DBI	euclidean	brute	8	4	0.32574
	Silhouette	euclidean	brute	8	4	0.78028
	Purity	l1	brute*	2	851	0.50645
Splice	ARI	l1	brute*	2	280	0.18572
	DBI	euclidean	brute*	4	33	0.96019
	Silhouette	euclidean	brute*	4	33	0.44728
	Purity	l1	brute*	2	280	0.72688
Vowel	ARI	euclidean	brute	4	154	0.08904
	DBI	euclidean	brute*	2	288	0.61628
	Silhouette	chebyshev	brute*	14	2	0.60045
	Purity	l1	brute*	2	281	0.93232

Table 1: Best configurations for each dataset and metric using the OPTICS algorithm. Algorithms marked with * achieved identical scores for *brute*, *ball_tree*, and *kd_tree*.

Table 1 illustrates the optimal parameter configurations identified for each dataset and metric, alongside the number of clusters detected and the corresponding metric scores.

For the **Satimage** dataset, we obtained a very low ARI of 0.02208; indicating that the clusters found by OPTICS do not correspond well to the actual classes. The low DBI (1.44691) suggests that clusters are compact and well-separated. The Silhouette score is high (0.78028), indicating that the clusters are well-separated and are cohesive internally. A Purity of 0.50645 means that about half of the data points are correctly clustered according to the most frequent class in each cluster.

For **Splice**, we obtained a higher ARI of 0.18572; indicating some agreement between the clustering and the true labels. The DBI is slightly better with a score of 1.30015, indicating more compact and better separated clusters. The Silhouette decreased to 0.44728, implying that the clusters are less defined and there may be more overlaps. The Purity increased to 0.72688, a relatively high score indicating good clustering quality.

For **Vowel**, we obtained a very low ARI of 0.08904. DBI obtained its lowest value of 1.02959, indicating the best cluster compactness and separation. The Silhouette score of 0.60045 suggests moderately defined and clearly separated clusters. Purity obtained its highest value of 0.93232, showing an excellent clustering quality.

Overall, the predominant optimal value for *metric* across these configurations is *euclidean*, but there appears to be a tendency for internal metrics to favor $l1$. All configurations selected $algorithm = brute$; *brute* computes all pairwise distances, which can be computationally expensive but can also be more accurate. In some cases (marked with *), *ball_tree*, *kd_tree*, and *brute* produced identical scores. This may occur because OPTICS overrides the *algorithm* parameter to *brute* for sparse datasets, which is the case for Splice and Vowel. The number of *min_samples* also varies. For external metrics

like ARI and Purity, low *min_samples* values often produce many small clusters, increasing alignment with true labels which benefits external metrics. For internal metrics, higher *min_samples* can lead to fewer, more cohesive and well-defined clusters which benefits internal metrics.

It is noteworthy that external metrics (ARI and Purity), which assess alignment with true labels, often correspond to a large number of clusters; whereas internal metrics (DBI and Silhouette), which emphasize cluster cohesion and separation, typically result in fewer clusters. For example, Satimage shows a harsh contrast with 4 clusters for DBI and Silhouette versus hundreds for ARI and Purity. However, there are inconsistencies, such as DBI producing the highest number of clusters (288) for the Vowel dataset. This indicates that the choice of metric influences whether clustering prioritizes internal structure or alignment with actual class labels, and the number of clusters also depends on the dataset.

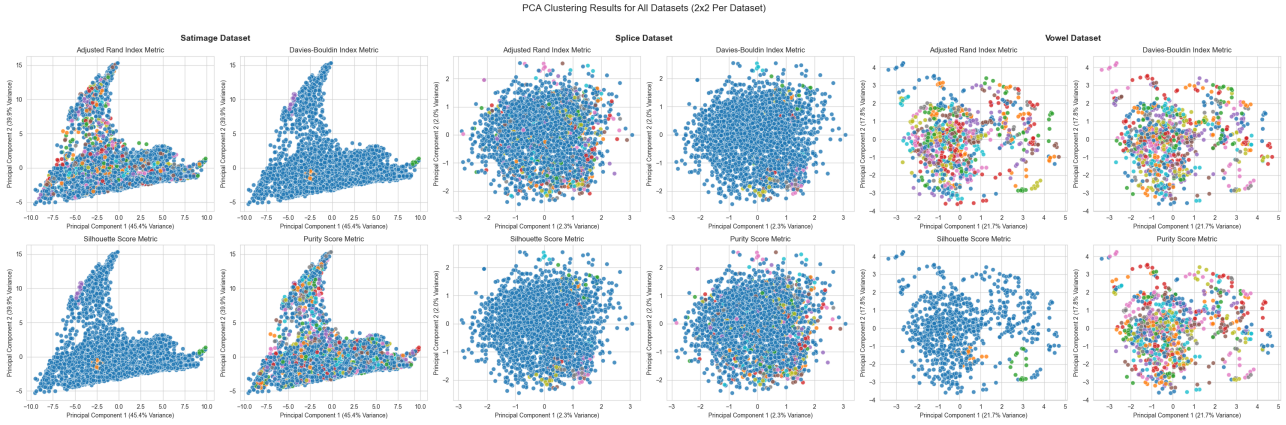

Figure 2: PCA visualizations of clusters for Satimage, Splice, and Vowel datasets using the best OPTICS configurations for each metric. Most points are classified as noise (blue). Variance percentages for each Principal Component are in brackets.

Figure 2 illustrates the clusters identified by the optimal OPTICS configurations within a PCA-reduced feature space. Overall, a large proportion of data points were labeled as noise. Internal metrics identified fewer but more cohesive clusters, while external metrics identified more clusters with sparser data points.

For **Satimage**, ARI and Purity produced a large number of sparse clusters that overlapped with noise. In contrast, DBI and Silhouette were able to identify identical cohesive clusters, which might capture relevant characteristics of the data. Table 1 shows that both metrics achieved the same optimal configuration.

For **Splice**, ARI and Purity also produced a large number of sparse clusters, whereas DBI and Silhouette identified a smaller number of compact and cohesive clusters. In this case, both internal and both external metrics obtained the same optimal configuration and identified the same clusters.

For **Vowel**, DBI produced a large number of clusters, unlike the other datasets. However, Silhouette continued to identify few cohesive clusters. ARI and Purity also identified a large number of sparse clusters.

In conclusion, these visuals provide a clearer representation of the clustering produced by OPTICS, demonstrating that the optimal configurations found by Silhouette yield the most cohesive and well-separated clusters. However, OPTICS does not seem to effectively separate the clusters, as some of the next algorithms offer higher quality clusterings, highlighting the limitations of OPTICS for these datasets.

For the OPTICS algorithm, we will omit confusion matrix graphics as they do not provide addi-

tional insights beyond those conveyed by the PCA visualizations. Since the majority of data points were classified as noise, the confusion matrices for all datasets and metrics show that nearly all data points belong to the noise class. Additionally, the vast number of clusters makes these confusion matrices difficult to interpret.

5 Spectral Clustering

In this section, we explore the application of the Spectral Clustering algorithm proposed by Ng et al. [5] and Von Luxburg [6].

5.1 Experiments and Results

We evaluated the performance of the Spectral Clustering algorithm on the Satimage, Splice, and Vowel datasets. The parameters explored were $n\_neighbors \in \{3, 5, 10, 15, 20, 25\}$, $affinity \in \{'nearest\_neighbors', 'rbf'\}$, $eigen\_solver \in \{'arpack', 'lobpcg', 'amg'\}$, $assign\_labels \in \{'kmeans', 'cluster\_qr'\}$, and $n\_clusters \in [2, 15]$. We explored additional parameters and more thoroughly than specified to achieve more comprehensive results.

N_neighbors sets the number of neighbors for constructing the affinity matrix. *Affinity* determines the method for measuring similarity between points. *Eigen_solver* selects the algorithm for eigenvalue decomposition. *Assign_labels* specifies the strategy for labeling clusters in the embedding space. *N_clusters* sets the dimension of the projection subspace.

Dataset	**Metric**	*n_neighbors*	*affinity*	*eigen_solver*	*assign_labels*	*n_clusters*	**Metric Score**
Satimage	ARI	10	nearest_neighbors	lobpcg	cluster_qr	4	0.63288
	DBI	3	nearest_neighbors	amg	kmeans	2	0.63789
	Silhouette	NA	rbf	amg	kmeans	2	0.49307
	Purity	30	nearest_neighbors	amg	cluster_qr	12	0.83714
Splice	ARI	50	nearest_neighbors	amg	kmeans	4	0.40496
	DBI	3	nearest_neighbors	amg	kmeans	3	2.32173
	Silhouette	50	nearest_neighbors	amg	cluster_qr	2	0.01292
	Purity	30	nearest_neighbors	arpack	kmeans	6	0.84065
Vowel	ARI	50	nearest_neighbors	amg	kmeans	12	0.14730
	DBI	50	nearest_neighbors	arpack	kmeans	15	1.65932
	Silhouette	50	nearest_neighbors	arpack	kmeans	15	0.16963
	Purity	50	nearest_neighbors	amg	kmeans	15	0.35152

Table 2: Best configurations for each dataset and metric using the Spectral Clustering algorithm.

Table 2 displays the optimal parameter configurations identified for each dataset and metric, alongside the corresponding metric scores.

For the **Satimage** dataset, we obtained a moderate ARI of 0.63288; indicating that clusters found by Spectral Clustering correspond reasonably well to the actual classes in the data. The DBI of 0.63789 is low and suggests that clusters are compact and well-separated. The Silhouette score is moderate with 0.49307, suggesting that clusters are fairly separated and are cohesive internally. A Purity of 0.83714 indicates that most of the data points are correctly clustered according to the most frequent class in each cluster.

For **Splice**, we obtained a lower ARI of 0.40496; indicating less agreement between the clustering and the true labels. The DBI is much higher with a score of 2.32173, indicating more sparse and poorly separated clusters. The Silhouette significantly decreased to 0.01292; implying that the clusters are not well-defined and near cluster boundaries. The Purity obtained a similar value than before with 0.84065, indicating that the clustering was effective.

For **Vowel**, we obtained the lowest ARI of 0.14730, suggesting the worst correspondence between clusters and actual classes. DBI score is 1.65932 which indicates partially sparse and poorly separated

clusters. The Silhouette score of 0.16963 shows poorly defined clusters and near cluster boundaries. Purity obtained its lowest value of 0.35152, suggesting a low clustering quality.

Overall, there is a tendency to use a higher number of neighbors for larger datasets, as observed with Splice and Vowel. This makes sense because these datasets contain more data points, requiring consideration of a larger number of neighbors to identify relevant clusters effectively. The predominant optimal value for $affinity$ is $nearest\_neighbors$, which is suitable for capturing the local structure of the data. For $eigen\_solver$, amg is the most frequently used option, followed by $arpack$ used occasionally, and $lobpcg$ used only once. Regarding $assign\_labels$, $kmeans$ is the most commonly selected method, while $cluster\_qr$ is used only in three configurations. The number of clusters chosen for each case generally aligns closely with the actual number of classes. However, there is an exception in the Satimage dataset, where the best Purity value is achieved with 12 clusters, which is double the number of actual classes.

Figure 3: PCA visualizations of clusters for Satimage, Splice, and Vowel datasets using the best Spectral Clustering configurations for each metric. Variance percentages for each Principal Component are in brackets.

Figure 3 visualizes the clusters identified by the optimal Spectral Clustering configurations within a PCA-reduced feature space.

For **Satimage**, ARI ($n\_clusters = 4$) produced well-separated clusters closely matching the true class count (6), although there is some overlap among orange, red, and blue data points. DBI ($n\_clusters = 2$) grouped most of the data points into a single cluster, while Silhouette Score ($n\_clusters = 2$) resulted in more data points in the minor cluster and better separation. Purity ($n\_clusters = 12$) created finer partitions, as evidenced by a higher number of smaller clusters with some overlap.

For **Splice**, ARI ($n\_clusters = 4$) generated four distinct clusters that overlap at the center, similar to the clustering produced by Purity ($n\_clusters = 6$) but included two additional minor clusters—one at the center and another at the bottom. DBI ($n\_clusters = 6$) grouped most data points into a single cluster and included very small minor clusters at the bottom. Silhouette ($n\_clusters = 2$) separated data points into two clusters with some central overlap.

For **Vowel**, all metrics produced a high number of clusters close to the true class count (11). The generated clusters exhibit significant overlap, especially in the central area, and many clusters have points scattered throughout the feature space.

Overall, ARI and Purity metrics produced higher-quality clustering with better cohesion and separation between clusters. Purity preferred to divide larger clusters into smaller ones, as seen in the Satimage and Splice examples.

Figure 4 presents the normalized confusion matrices comparing predicted clusters to true labels

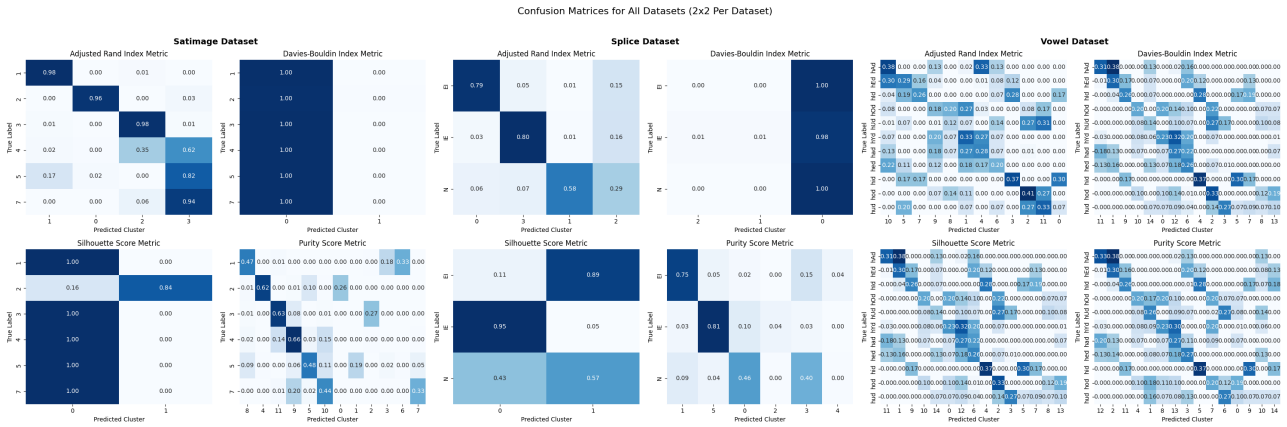

Figure 4: Confusion matrices of Spectral Clustering for each dataset across different metrics, normalized row-wise, comparing predicted clusters to true labels.

across the evaluated datasets and clustering metrics.

For **Satimage**, the confusion matrices show varying degrees of alignment between predicted clusters and true classes, depending on the metric used. Using ARI, certain clusters correspond to specific classes but with notable overlaps. For example, Class '4' is split between Cluster '2' (36%) and Cluster '3' (64%); while Cluster '3' is distributed among Class '4' (26%), Class '5' (34%), and Class '7' (40%). This indicates partial alignment with significant fragmentation. In contrast, DBI and Silhouette Score result in fewer clusters, leading to broader groupings where almost all classes are merged into a single Cluster. The Purity Score introduces more clusters, increasing granularity but also dispersion, reflecting a trade-off between cluster purity and class correspondence.

For **Splice**, the ARI confusion matrix exhibits strong alignment with high diagonal values—Class 'EI' aligns with Cluster '0' at 90%, Class 'IE' with Cluster '3' at 87%, and Class 'N' primarily with Cluster '0' at 58% and Cluster '2' at 29%. This indicates effective clustering corresponding to true classes. However, DBI classifies all Classes into Cluster '0', indicating poor clustering quality. The Silhouette and Purity scores present good alignment for Classes 'EI' and 'IE', which are defined within single clusters, but Class 'N' is divided into multiple clusters.

For **Vowel**, the high number of clusters makes it complex to analyze these confusion matrices. High diagonal values are observed across all metrics, indicating moderately high alignment between clusters and true labels. However, no cluster exclusively represents a single class; every class is distributed among multiple clusters due to the large number of finer clusters created. This suggests an overlap between several classes within clusters, as depicted in the corresponding PCA Figure 3.

6 K-Means

The K-Means method seeks to divide a set of n data points into k clusters. The algorithm tries to minimize the average distance between the points classified into the same cluster [7].

6.1 Variant: Global K-Means (GKM)

The global K-Means [8] variation addresses the limitation of high sensitivity to cluster initialization. It dynamically adds one cluster center at a time: from all the possible positions for the new cluster center, the one that minimizes the clustering error is selected as the optimal new center. The exhaustive search significantly increases the computational cost of the algorithm. However, as global K-Means is deterministic, it needs only a single execution.

6.2 Experiments and Results

Distance metrics can capture distinct types of relationships between samples, based on the nature of the data. Therefore, three different distance metrics were employed: *euclidean*, *manhattan* and *cosine*.

A convergence threshold has been defined to ensure a minimal change in the centroid positions between iterations. This approach makes the algorithm halt when improvement is practically insignificant, enhancing efficiency. The tolerance was set to 10^{-5} after empiric experimentation, achieving a trade-off between precision and computational time. In the scope of the same topic, the maximum number of iterations is set to 100. We could observe that convergence typically occurred within 30-50 iterations, making 100 iterations adequate to achieve convergence for most datasets.

The performance for both K-Means and Global K-Means is evaluated using a range of clusters $k \in [2, 15]$ and $distance \in \{euclidean, manhattan, cosine\}$.

6.2.1 Experiment 1: Metric-Specific Analysis

Table 3 presents the best-performing configurations identified for both algorithms. For visual clarity, configurations from Global K-Means that match those of K-Means are indicated with a dash ('-').

Dataset	**Metric**	**K-Means**			**Global K-Means**		
		k	*distance*	**Metric Score**	k	*distance*	**Metric Score (Improvement)**
Satimage	Adjusted Rand Index	6	cosine	0.5798	10	cosine	0.6000 (+0.0202)
	Davies-Bouldin Index	3	manhattan	0.8118	–	–	–
	Silhouette Score	3	euclidean	0.4373	–	–	–
	Purity Score	14	cosine	0.8321	–	–	–
Splice	Adjusted Rand Index	4	euclidean	0.4305	4	cosine	0.3915 (-0.039)
	Davies-Bouldin Index	15	manhattan	6.5208	14	euclidean	1.4700 (-5.051)
	Silhouette Score	2	manhattan	0.01727	3	manhattan	0.0141 (-0.0032)
	Purity Score	10	cosine	0.9005	14	manhattan	0.8054 (-0.0951)
Vowel	Adjusted Rand Index	6	euclidean	0.1708	6	cosine	0.1537 (-0.0171)
	Davies-Bouldin Index	15	manhattan	1.6468	15	euclidean	1.5675 (-0.0793)
	Silhouette Score	2	euclidean	0.1809	–	–	–
	Purity Score	14	euclidean	0.3585	15	euclidean	0.3434 (-0.0151)

Table 3: Best configurations for each dataset and metric using K-Means and Global K-Means algorithms, along with their respective score.

No single configuration (number of clusters and distance metric) performs universally well across all datasets or evaluation metrics. Both algorithms achieve comparable performance. For the **Satimage** dataset, the configurations are nearly identical for both algorithms, while for **Splice** and **Vowel**, GKM yields always slightly lower scores compared to traditional K-Means. This suggest that GKM deterministic approach is not suitable for data with high-dimensional spaces. Although the differences are not large, they still represent a minor performance downgrade. Given the near-identical clustering divisions, only the plots from the K-Means results will be shown in the analysis, to avoid redundancy.

The evaluation of the best model for the **Satimage** dataset reveals differing results depending on internal and external indexes. DBI and Silhouette Score suggest that fewer clusters ($k = 3$) offer better compactness and separation, particularly with *manhattan* and *euclidean* distances yielding the best performance, respectively. However, Purity indicates that a higher number of clusters ($k = 14$) aligns better with the ground truth classes, achieving a score of 0.8321 using *cosine* distance. This suggests that the data may contain subclasses within broader categories, with *cosine* distance being effective in capturing angular relationships between data.

For the **Splice** dataset, the metrics show varied outcomes. Purity achieves the highest score (0.9005) with $k = 10$ clusters using *cosine* distance. Meanwhile, ARI is optimal at $k = 4$ (0.4305) using *euclidean* distance, slightly above the true class count of 3. Silhouette Score peaks at $k = 2$ but reflects poor clustering separation with a low value of 0.01727, using *manhattan* distance. DBI, optimized at $k = 15$ with a score of 6.5208 using *manhattan* distance, also indicates significant overlap in the class space. Overall, the metrics suggest that the data structure has overlapping class boundaries.

The **Vowel** dataset, characterized by a high number of classes (11), shows weak alignment with the true classes across metrics. ARI (0.1708) and Purity (0.3585) remain low regardless of cluster count, with Purity optimized at $k = 14$ using *euclidean* distance. DBI is optimized at $k = 15$ (1.6468) with *manhattan* distance, indicating better internal compactness. However, Silhouette Score (0.1809), even at $k = 2$ using *euclidean* distance, underscores poor clustering separation. These results suggest that the dataset has a complex structure, likely due to high intra-class variability or significant class overlap, as even with $k = 15$, alignment with true classes remains weak.

6.2.2 Experiment 2: PCA-Based Cluster Visualization

Figure 5: PCA visualization of clusters for Satimage, Splice, and Vowel datasets using the best configurations for each metric.

In Figure 5, we can observe that for **Satimage**, the ARI and Purity metrics resulted in more clusters with diverse shapes, compared to DBI and Silhouette. This makes sense, as DBI and Silhouette aim to create well-shaped, compact clusters. Additionally, we can see that the only PCA plot with overlapping clusters is the one using the Purity metric. This could be due to the larger number of clusters used, compared to the other metrics.

For **Splice**, it is noteworthy that ARI produced better clusters than DBI in terms of compactness and spacing, which is unexpected. This discrepancy can be attributed to the fact that the variance explained by the PCA is only 2.3%, suggesting that the data structure we are observing differs substantially from the one in the true-dimensional space. Additionally, the other metric that assesses the structure of the clusters, namely the Silhouette score, only considers two clusters.

For **Vowel**, a similar situation to that of the **Splice** dataset arises, where it appears that ARI produces better clusters than DBI. However, in this case, the issue can be attributed to the fact that DBI is also attempting to create clusters with more uniform structures. This discrepancy is further explained by the fact that the variance explained is nearly ten times greater than that of the **Splice** dataset.

In conclusion, these visuals offer a clearer representation of the clustering produced by K-Means.

Additionally, they highlight some of the limitations of using this technique, particularly when the variance explained is low, which can lead to distorted or incomplete cluster structures.

6.2.3 Experiment 3: Confusion Matrix Analysis

In Figure 6 for **Satimage**, some of the metrics demonstrate good alignment between the predicted clusters and the true labels. One such metric is ARI, where we can observe that certain clusters, like Cluster 5 with True Label 3, are highly accurate, with 97% of the instances correctly predicted. However, there are also cases of confusion, such as Label 5, which in our predictions is merged between clusters 0 and 3. Purity also shows a good alignment overall, but its precision is relatively poor, indicating that while the majority of instances are correctly grouped, there are significant inaccuracies within the clusters.

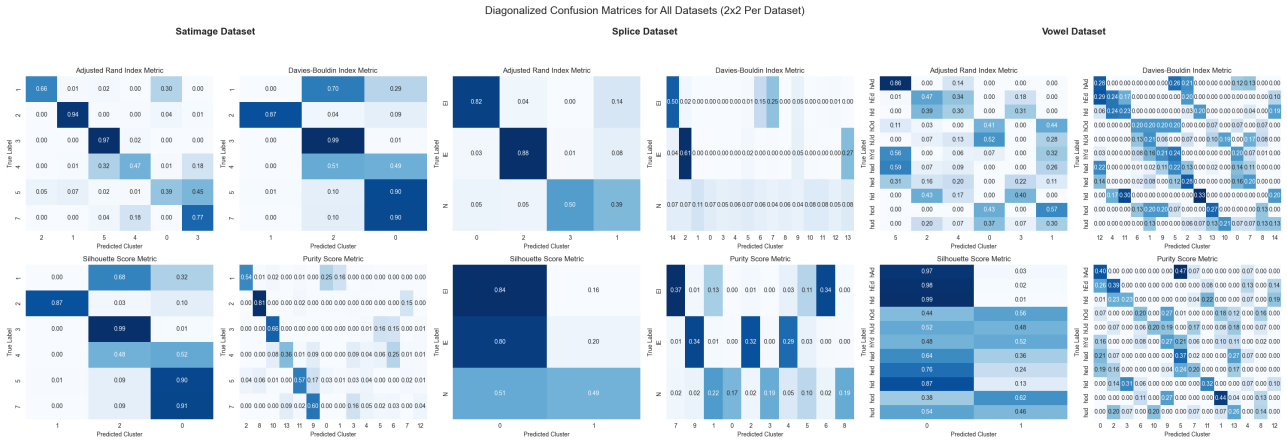

Figure 6: Confusion matrices for Satimage, Splice, and Vowel datasets across different metrics, normalized row-wise, comparing predicted clusters to true labels.

For **Splice**, the only metric that shows relatively good alignment is ARI. We can observe that the true labels *IE* and *EI* are very well represented, with more than 80% of the instances correctly classified. However, the label *N* is primarily spread across two clusters, Cluster 3 and Cluster 1, leading to a significant amount of misclassification for this particular label.

For **Vowel**, the high number of clusters makes the confusion matrix difficult to analyze, as it becomes increasingly complex with more clusters. However, despite this complexity, the Purity score shows a somewhat well-diagonalized matrix, indicating some alignment between predicted clusters and true labels. However, the maximum accuracy is only 44%, which is low given the large number of clusters. This can be attributed to the nature of K-Means, where many points may be assigned to clusters other than their true label due to the proximity to centroids, making it easier for points to be misclassified.

7 X-Means

One of the biggest issues related to the use of the K-Means algorithm is to find the best k to use in the dataset, therefore algorithms such as X-Means [9] have been developed to address this issue.

Key aspects of the algorithm's implementation are as follows: as outlined in the paper, the division of the centroid must be proportional to the size of the cluster. To achieve this, the distance of the cluster is calculated as the maximum Euclidean distance between any point and the centroid. A random direction is then selected, and the two new centroids are positioned along this direction at a distance equal to half of the computed maximum distance. This method is designed to position the

new centroids between the midpoint and the boundary of the cluster, facilitating their convergence within the cluster during subsequent iterations.

Another important consideration is the computation of the Bayesian Information Criterion (BIC). In certain cases, the variance can be zero, which would prevent the computation of the logarithm of the variance, a necessary step for calculating the BIC. To address this issue, we assign a very small value 10^{-6} to the variance in such cases, ensuring the computation can proceed without errors.

7.1 Experiments and Results

We evaluate the performance of the X-Means algorithm by exploring its sole hyperparameter, k_max. The range tested is $k\_max \in [4, 8, 16, 32, 64, 128, 256, 512, 1024]$ and each experiment is run over 5 seeds. This repetition is necessary because the X-Means algorithm introduces randomness in the splitting direction, therefore we will choose the better seed for every score.

Additionally, we do not experiment with alternative distance metrics, as the algorithm inherently employs Euclidean distance to calculate cluster sizes and perform proportional splits. Consequently, the K-Means algorithm used inside X-Means also utilizes Euclidean distance and maintains the same number of iterations as specified in the experiments above, which is set to 100. As noted, the sole hyperparameter under consideration is k_max.

7.1.1 Experiment 1: Metric-Specific Analysis

In Table 7, we present the results of the experiments with the best metrics achieved and their corresponding hyperparameters.

Dataset	Metric	k_max	Best k	Best Score
Satimage	Adjusted Rand Index	8	8	0.5575
	Davies-Bouldin Index	4	4	1.1758
	Silhouette Score	4	4	0.3934
	Purity Score	4	4	0.8253
Splice	Adjusted Rand Index	4	4	0.3084
	Davies-Bouldin Index	4	4	7.3241
	Silhouette Score	4	4	0.0118
	Purity Score	4	4	0.6354
Vowel	Adjusted Rand Index	8	8	0.1738
	Davies-Bouldin Index	1024	586	0.4195
	Silhouette Score	256	256	0.4063
	Purity Score	4	4	0.5859

Figure 7: Best configurations for each dataset and metric using the X-Means algorithm.

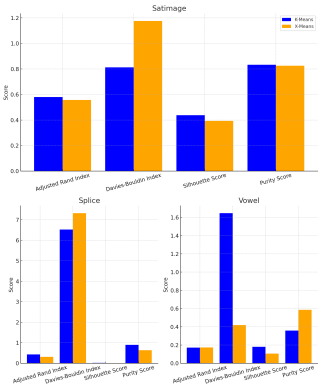

Figure 8: Comparative K-Means vs X-Means (size of graph is for visualisation purposes)

Comparing K-Means and X-Means, see Figure 8, it is evident that for Satimage and Splice, K-Means outperforms X-Means across all metrics. However, this trend is reversed for Vowel, where X-Means demonstrates superior performance in ARI (0.1738 vs. 0.1708), DBI (0.4195 vs. 1.6468), and Purity Score (0.5859 vs. 0.3585). With these experiments, it seems that in scenarios where the best k found by X-Means falls within the range explored by the K-Means grid search, K-Means outperforms X-Means. This can be attributed to the fundamental differences in how the two algorithms determine the optimal number of clusters. In K-Means, a grid search is performed over a set of k values, and the best k is selected based on the metric that provides the highest score. In contrast, X-Means determines the optimal k by optimizing BIC, rather than directly optimizing the metric being evaluated. This approach leads to an inherent limitation in X-Means, causing it to underperform in these scenarios compared to K-Means.

7.1.2 Experiment 2: PCA-Based Cluster Visualization

Figure 9: PCA visualization of clusters for Satimage, Splice, and Vowel datasets using the best configurations for each metric.

As can be seen in Figure 9, this deviates from the results found in Figure 5 using K-Means. None of the clusters identified by X-Means are identical to those produced by K-Means. However, in the **Splice Dataset**, we observe that some clusters are either equal or overlap significantly, indicating that they maximize the metric scores in the experiment. Additionally, as expected, in some of the **Vowel** graphics, it is evident that defined clusters are not visible due to the very large k values.

In conclusion, we see that X-Means produces very different results compared to K-Means. Additionally, X-Means can sometimes identify very large values of k, which, when projected into lower dimensions, may lack discernible cluster structure, as observed in **Vowel**. However, in other scenarios, such as **Satimage**, it demonstrates the ability to produce well-defined clusters even in lower dimensions, showcasing its potential for effective clustering under certain conditions.

7.1.3 Experiment 3: Confusion Matrix Analysis

Figure 10: Confusion matrices for Satimage, Splice datasets across different metrics, normalized row-wise, comparing predicted clusters to true labels.

In this case, only **Satimage** and **Splice** are shown, as the results for **Vowel** are either unreadable due to the very large k values or are not insightful. As expected from the previous results, some of the confusion matrices are identical, all of them within the **Splice** dataset. Surprisingly, all of

the confusion matrices exhibit a more or less well-defined diagonal compared to the K-Means results, indicating that each cluster captures a real label quite effectively.

8 Fuzzy Clustering

In this section, we explore the application of the Generalised Suppressed Fuzzy C-Means (gs-FCM) clustering algorithm, specifically using the gsξ-FCM suppression scheme as proposed by Szilágyi and Szilágyi [10], which improves the the standard Fuzzy C-Means (FCM) algorithm by introducing a suppression mechanism.

The key implementation decisions were:

1. **Membership Initialization:** The membership matrix U is randomly initialized using a Dirichlet distribution, ensuring each row sums to one.
2. **Two-Step Membership Update:** Standard FCM membership values are first computed, then suppressed using $\mu_w = (\sin(\frac{\pi u_w}{2}))^{\xi}$ for the winner membership and proportional adjustment for the others.
3. **Normalization and Numerical Stability:** Memberships are renormalized after suppression, and small constants (e.g., 10^{-10}) prevent division by zero.
4. **Convergence Criteria:** Iterations stop when changes in U fall below a given threshold or when the maximum iteration count is reached.

8.1 Experiments and Results

In this section, we evaluate the performance of the gsξ-FCM algorithm on the Satimage, Splice, and Vowel datasets. Clusters were tested with $k \in [2, 15]$, fuzziness parameter $m \in \{1.5, 2.5, 3.5\}$, and suppression parameter $\xi \in \{0.3, 0.5, 0.7\}$. The maximum number of iterations was capped at 100.

8.1.1 Experiment 1: Metric-Specific Analysis

Table 4 summarizes the best configurations for each dataset and metric. Each configuration demonstrates how gsξ-FCM adapts to specific dataset characteristics.

For the **Satimage dataset**, ARI peaks (0.565) at $k = 7$, $m = 1.5$, $\xi = 0.3$, marginally exceeding the true six classes to capture additional variation. Both DBI (0.811) and Silhouette Score (0.437) favor $k = 3$, though with differing fuzziness ($m = 3.5$ and $m = 2.5$ respectively), indicating compact clusters with varying boundary definitions. Purity maximizes (0.827) at $k = 15$, $m = 2.5$, $\xi = 0.5$, where finer partitioning and moderate suppression capture detail. While metric-specific preferences vary, $k = 3$, $m = 2.5$, and $\xi = 0.3$ emerge as prevalent values.

For **Splice**, ARI achieves its maximum (0.593) at $k = 3$, matching the true classes, with high fuzziness ($m = 3.5$) and low suppression ($\xi = 0.3$). DBI minimizes (5.825) at $k = 15$,

Dataset	Metric	k	m	ξ	Seed	Value
Satimage	ARI	7	1.5	0.3	0	0.565
	DBI	3	3.5	0.7	3	0.811
	Silhouette	3	2.5	0.3	0	0.437
	Purity	15	2.5	0.5	2	0.827
Splice	ARI	3	3.5	0.3	2	0.593
	DBI	15	1.5	0.7	3	5.825
	Silhouette	2	2.5	0.5	0	0.016
	Purity	8	1.5	0.3	4	0.920
Vowel	ARI	13	1.5	0.5	2	0.186
	DBI	14	2.5	0.7	2	1.585
	Silhouette	2	3.5	0.5	0	0.183
	Purity	15	2.5	0.7	1	0.387

Table 4: Best configurations for each dataset and metric using the gsξ-FCM algorithm, along with their respective metric values and seeds.

$m = 1.5$, $\xi = 0.7$, favoring compact clusters, while Silhouette Score's minimum (0.016) at $k = 2$ suggests oversimplification. Purity peaks (0.920) at $k = 8$, $m = 1.5$, $\xi = 0.3$, revealing latent structures through moderate suppression. Parameter preferences show significant metric-dependent variation.

For **Vowel**, ARI peaks (0.186) near the true 11 classes at $k = 13$, $m = 1.5$, $\xi = 0.5$, indicating substantial class overlap. DBI improves (1.585) with flexible boundaries at $k = 14$, $m = 2.5$, $\xi = 0.7$, while Silhouette Score maximizes (0.183) with fewer, fuzzier clusters. Purity (0.387) favors maximum granularity ($k = 15$, $m = 2.5$, $\xi = 0.7$). The dataset generally benefits from higher parameter values, with $m = 1.5$ suitable for true class counts and $m = 2.5$ for handling increased complexity.

The following section presents PCA projections to visualize cluster structures and evaluate metric-specific results.

8.1.2 Experiment 2: PCA-Based Cluster Visualization

Figure 11 visualizes clusters from the best configurations for each metric (Table 4) in PCA-reduced space.

For **Satimage**, ARI ($k = 7$) produces well-separated clusters closely matching the true class count (6). DBI ($k = 3$) forms three compact clusters prioritizing separation over granularity, while Silhouette Score ($k = 3$) yields similar groups with *slightly* sharper boundaries. Purity ($k = 14$) creates finer partitions, evident in the higher number of smaller clusters with some overlap.

In **Splice**, ARI ($k = 3$) shows moderately distinct clusters consistent with the true class count (3). DBI ($k = 15$) results in fragmented, overlapping clusters, reflecting the effect of larger k. Silhouette Score ($k = 2$) oversimplifies, producing two broad groups with significant loss of structural detail. Purity ($k = 8$) captures latent subclasses with moderately distinct clusters. However, the low combined variance explained (4.33%) obscures much of the structure, limiting PCA's ability to represent separability.

For **Vowel**, ARI ($k = 10$) captures partial separation with significant overlap, expected given the dataset's complexity. DBI and Purity ($k = 15$) create finer clusters, reducing overlap but failing to fully resolve class structures. Silhouette Score ($k = 2$) simplifies excessively, producing two poorly-separated groups. Once again, the PCA's two principal components do not represent much variance, so the readings we can do on these plots are limited.

Figure 11: PCA visualization of clusters for Satimage, Splice, and Vowel datasets using the best configurations for each metric.

8.1.3 Experiment 3: Confusion Matrix Analysis

Figure 12 displays normalized confusion matrices comparing predicted clusters against true labels across datasets and clustering metrics.

For **Satimage**, the ARI metric achieves high accuracy with a prominent diagonal pattern, though it creates an additional cluster capturing variance from Classes 1, 2, and 5. DBI and Silhouette metrics produce fewer clusters, merging multiple classes (notably Classes "5" and "7"). The Purity Score increases granularity through additional clusters, maintaining strong diagonal relationships but dispersing classes across multiple clusters.

The **Splice** dataset exhibits strong class-cluster alignment under ARI (87% for "EI"-Cluster "1", 94% for "IE"-Cluster "2", 80% for "N"-Cluster "0"). DBI and Purity Score generate more clusters with diffused alignments, while Silhouette's two-cluster solution groups "IE" and "EI" together, separating "N", suggesting an underlying pattern.

The **Vowel** dataset demonstrates weaker clustering performance across metrics. While ARI, DBI, and Purity show modest diagonal patterns, Silhouette oversimplifies by assigning most classes to Cluster 0. Despite the dataset's higher complexity, consistent groupings emerge across metrics for certain class pairs ("hud"/"hod", "hEd"/"hAd", "hed"/"had"), indicating potential latent relationships

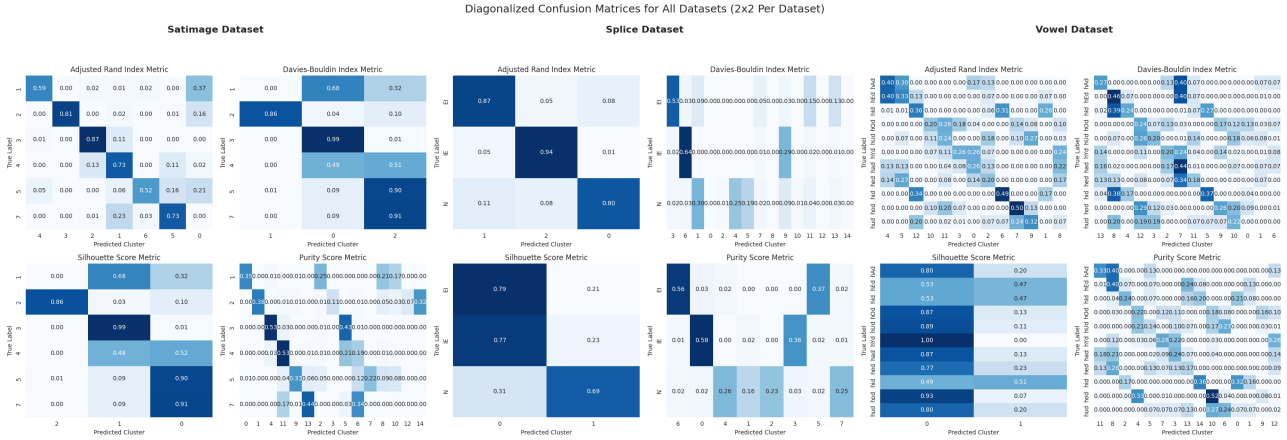

Figure 12: Confusion matrices for Satimage, Splice, and Vowel datasets across different metrics, normalized row-wise, comparing predicted clusters to true labels.

9 Comparison between algorithms

We compared clustering algorithms across the three datasets using the four metrics. Results are summarized in Figure 13 (averaged across datasets) and Figure 14 (split by dataset and metric). DBI was normalized using CDF for visualization.

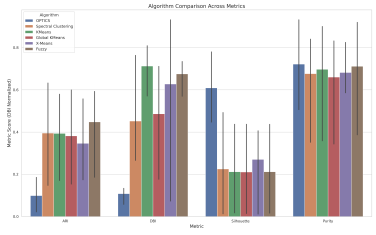

Figure 13: Average performance of clustering algorithms across metrics.

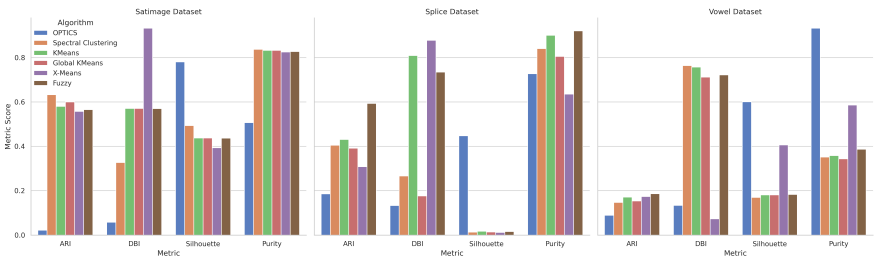

Figure 14: Performance of clustering algorithms split by dataset and metric.

OPTICS consistently identified well-separated clusters (highest Silhouette) and captured com-

pact internal cluster structures (lowest DBI). This behavior was prominent in datasets like Satimage, where clusters are naturally cohesive, and Vowel, where sparse distributions highlight OPTICS' adaptability. However, OPTICS' weak ARI and Purity scores suggest that while it models intrinsic cluster boundaries effectively, these may not align well with true class distributions, particularly in datasets like Splice where dense overlapping regions dominate.

Spectral Clustering and **KMeans**, on the other hand, excelled in datasets like Satimage and Splice where clear global structures dominate. Their superior ARI and Purity scores reflect an alignment with true labels, but lower Silhouette and DBI values indicate challenges in refining cluster definitions when classes overlap (as in Vowel). Spectral Clustering's ability to leverage non-linear embeddings allowed it to outperform KMeans on metrics like ARI for Satimage, indicating its edge in datasets with non-convex clusters.

Fuzzy Clustering, with its probabilistic assignment, uncovered nuanced patterns that distance-based algorithms struggled with. It excelled in Splice (top ARI and Purity), where overlapping regions benefit from its soft cluster boundaries, while maintaining competitive performance in Vowel. However, its slightly higher DBI compared to OPTICS and X-Means suggests less compact clusters, particularly in datasets with sparse regions.

X-Means excelled in compactness (DBI) across Satimage and Vowel, often outperforming KMeans due to its automatic determination of cluster counts. However, its ARI and Purity scores indicate that these clusters may not fully align with true labels, particularly in Splice, where its rigid reliance on spatial separability is less effective.

Global KMeans provided balanced results, with high Purity and ARI scores on Satimage and Splice, suggesting its ability to align with true global structures. However, its performance dropped on metrics like DBI and Silhouette for Vowel, highlighting its difficulty in adapting to sparse or overlapping clusters.

The dataset-specific analysis in Figure 14 reveals unique insights about each dataset. Satimage's cohesive cluster structure favors global alignment algorithms like Spectral Clustering and Global KMeans. Splice, with its dense and overlapping regions, challenges most algorithms but highlights the strength of probabilistic approaches like Fuzzy. Vowel, characterized by sparse and highly overlapping clusters, magnifies the trade-offs between compactness (OPTICS, X-Means) and label alignment (Spectral Clustering, Fuzzy).

10 Conclusions

In this work, we implemented and evaluated multiple clustering algorithms, including **OPTICS**, **Spectral Clustering**, **K-Means**, **X-Means**, **Global K-Means**, and a **Fuzzy Clustering** approach, across three datasets with varying characteristics. The analysis used both internal metrics, such as Davies–Bouldin Index (DBI) and Silhouette Coefficient, and external metrics, including Adjusted Rand Index (ARI) and Purity, to assess clustering performance.

The results demonstrate the importance of matching algorithms to the properties of the dataset and the evaluation metric. Internal metrics like DBI consistently favored **OPTICS**; however, this is due to the small number of clusters identified, which are highly cohesive, and the large number of points identified as noise. External metrics, such as ARI and Purity, revealed the strengths of **Spectral Clustering**, **K-Means**, and **Fuzzy Clustering**, which excelled at aligning cluster assignments with true labels, especially in datasets with overlapping or non-linear clusters.

The findings underscore that no single algorithm performs best across all scenarios. Instead, algorithm and parameter selection must be tailored to the dataset and the specific goals of the analysis. This comprehensive evaluation provides insights into the trade-offs between compactness, adaptability, and label alignment, offering guidance for selecting clustering algorithms in diverse contexts.

References

[1] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[2] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[3] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[4] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. page 49–60, 1999.

[5] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[6] Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[7] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[8] Aristidis Likas, Nikos Vlassis, and Jakob Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36:451–461, 08 2002.

[9] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[10] László Szilágyi and Sándor M. Szilágyi. Generalization rules for the suppressed fuzzy c-means clustering algorithm. *Neurocomputing*, 139:298–309, 2014.