SPECTRAL CLUSTERING AND ITS APPLICATIONS

BY OLGA GRIGORIEVA

Spectral Clustering

Spectral clustering is a graph-based clustering method that **transforms the data into a lower-dimensional space** using the eigenvectors of the graph Laplacian, then applies standard clustering (k-means) on the transformed data.

Implementation:

a. Construct a similarity graph using the input data.
b. Compute the graph Laplacian (un-normalized or normalized).
c. Perform eigendecomposition on the Laplacian.
d. Use the top eigenvectors to embed the data.
e. Cluster the embedded points.

Spectral methods can detect non-convex and non-linearly separable clusters, unlike classical clustering techniques like k-means.

Spectral Clustering algorithms

S. T. Wierzchoń and M. A. Kłopotek,
Modern Algorithms of Cluster Analysis

Normilized	This is a standard approach, uses the **normalised Laplacian**, often improving stability and interpretability.
Landmark	Designed for large-scale datasets, this method approximates the full similarity matrix using a subset of **"landmark"** points.
MNCut	This method minimises the normalized **cut criterion**.
Kernel	Generalizes spectral clustering by allowing **non-linear transformations** using **kernel** functions (RBF, polynomial).

Normalized Spectral Clustering

Normalized Laplacian: $L_{\text{norm}} = D^{-1/2} L D^{-1/2}$

where $L = D - W$, and D is the degree matrix, W is the similarity matrix.

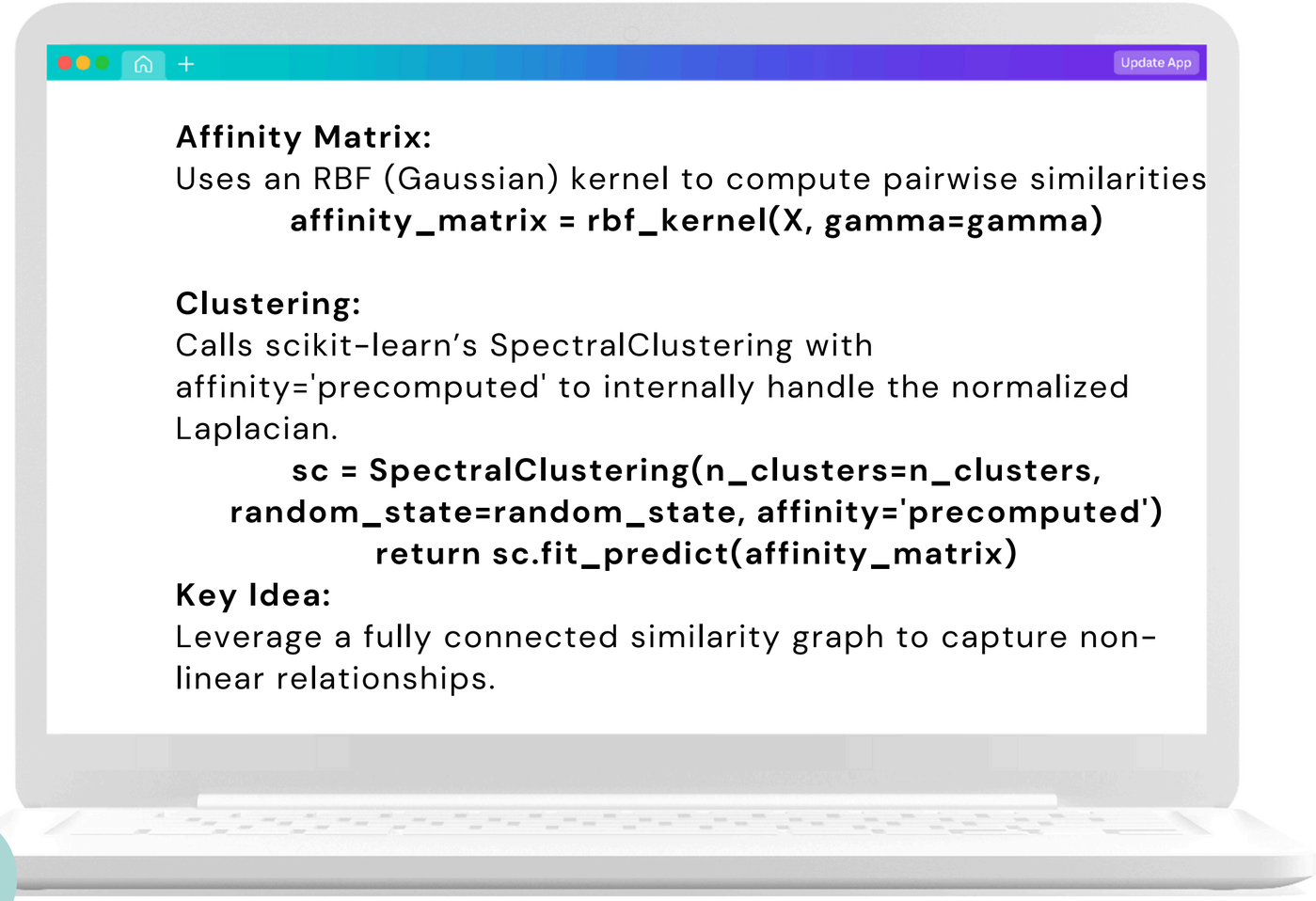
Advantages:

- Handles unbalanced clusters better.
- Ties to random walks on graphs.
- Commonly used in the Ng–Jordan–Weiss (NJW) algorithm.

Use case: Recommended when datasets have varying densities or cluster sizes.

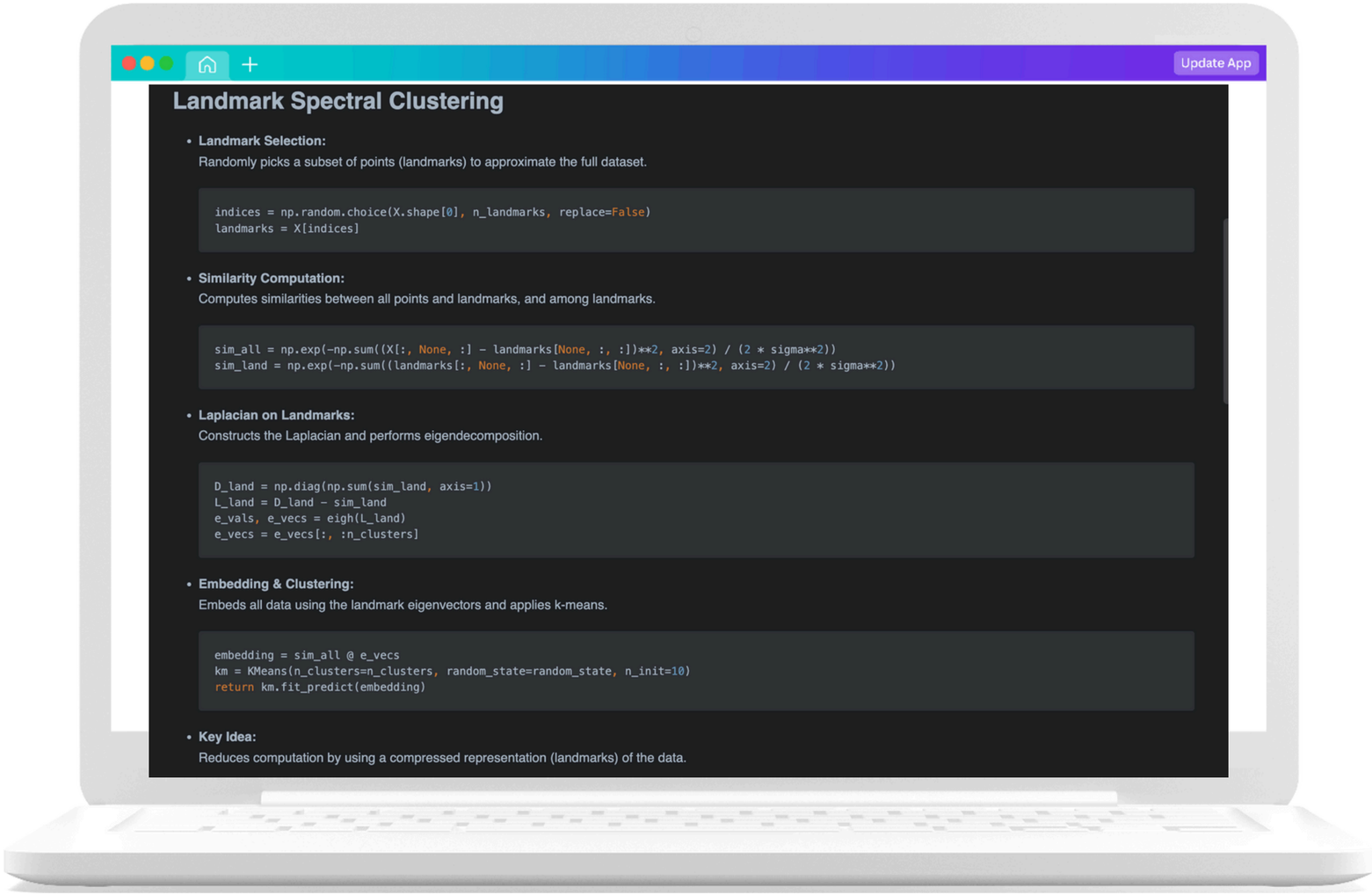
S. T. Wierzchoń and M. A. Kłopotek, Modern Algorithms of Cluster Analysis

Normalized Spectral Clustering



More implementations are in Readme

Readme


Update App
Landmark Spectral Clustering
Landmark Selection:
Randomly picks a subset of points (landmarks) to approximate the full dataset.
indices = np.random.choice(X.shape[0], n_landmarks, replace=False)
landmarks = X[indices]
Similarity Computation:
Computes similarities between all points and landmarks, and among landmarks.
sim_all = np.exp(-np.sum((X[:, None, :] - landmarks[None, :, :])**2, axis=2) / (2 * sigma**2))
sim_land = np.exp(-np.sum((landmarks[:, None, :] - landmarks[None, :, :])**2, axis=2) / (2 * sigma**2))
Laplacian on Landmarks:
Constructs the Laplacian and performs eigendecomposition.
D_land = np.diag(np.sum(sim_land, axis=1))
L_land = D_land - sim_land
e_vals, e_vecs = eigh(L_land)
e_vecs = e_vecs[:, :n_clusters]
Embedding & Clustering:
Embeds all data using the landmark eigenvectors and applies k-means.
embedding = sim_all @ e_vecs
km = KMeans(n_clusters=n_clusters, random_state=random_state, n_init=10)
return km.fit_predict(embedding)
Key Idea:
Reduces computation by using a compressed representation (landmarks) of the data.

Kernel Spectral Clustering

Replace similarity matrix with a kernel matrix computed in a high-dimensional space.

Steps:

a. Compute kernel matrix **K(xi,xj)K(xi,xj)**.
b. Normalize and compute the Laplacian.
c. Apply spectral embedding and cluster.

Advantages:

- Extremely flexible.
- Can separate clusters that are linearly inseparable in the original space.

S. T. Wierzchoń and M. A. Kłopotek,
Modern Algorithms of Cluster Analysis

Landmark Spectral Clustering

S. T. Wierzchoń and M. A. Kłopotek,
Modern Algorithms of Cluster Analysis

Designed for large-scale datasets, this method approximates the full similarity matrix using a subset of "landmark" points.

Steps:

a. Select $l \ll n$ representative landmarks.
b. Compute similarity between all data points and landmarks.
c. Build a low-rank approximation of the affinity matrix.
d. Perform spectral clustering on this compressed representation.

Strengths:

- Reduces computational complexity from $O(n^2)$ to $O(nl)$.
- Enables scalability to large datasets.

Trade-off:

- May lose some clustering precision depending on landmark selection quality.

MNCut Spectral Clustering

Minimum Normalized Cut

S. T. Wierzchoń and M. A. Kłopotek,
Modern Algorithms of Cluster Analysis

This method minimizes the normalized **cut criterion**:

$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(A,V)} + \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(B,V)}$$

Objective: Partition graph to minimize connections between clusters while keeping internal connectivity strong.

Implementation:

- Solves a generalized eigenvalue problem.
- Closely linked to random walks and Markov chains.

Advantages:

- Strong theoretical foundation.
- Well-suited for image segmentation and balanced partitioning tasks.

Methodology

Gagolewski's framework provides a suite of benchmark datasets with varying characteristics

1. Using the Adjusted Asymmetric Accuracy (AAA) as the primary evaluation metric, as suggested by Gagolewski.
2. The AAA measures the similarity between the obtained cluster assignments and the ground-truth (reference) partitions.
3. Calculating the AAA score for each run of each algorithm on each dataset.
4. For each dataset and algorithm, computing the mean and standard deviation of the AAA scores across the multiple runs.

Gagolewski M., A framework for benchmarking clustering algorithms, SoftwareX 20, 2022

Datasets

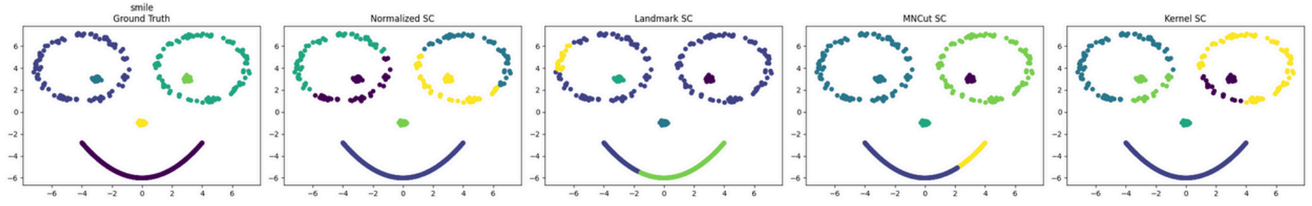
Benchmark Suite for Clustering Algorithms that consists of tree benchmark batteries (dataset collections). Each battery consists of several datasets of different origins.

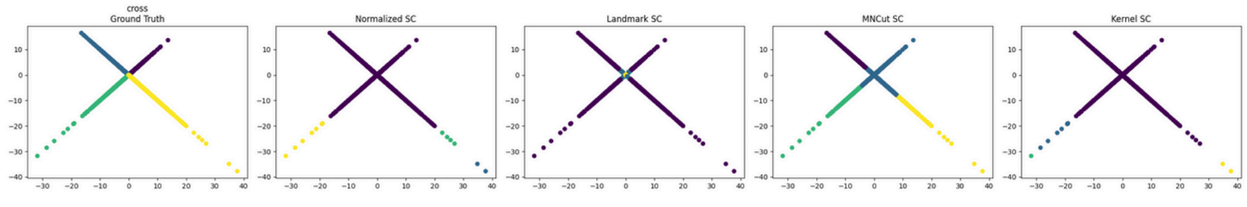
https://github.com/gagolews/cl ustering-data-v1

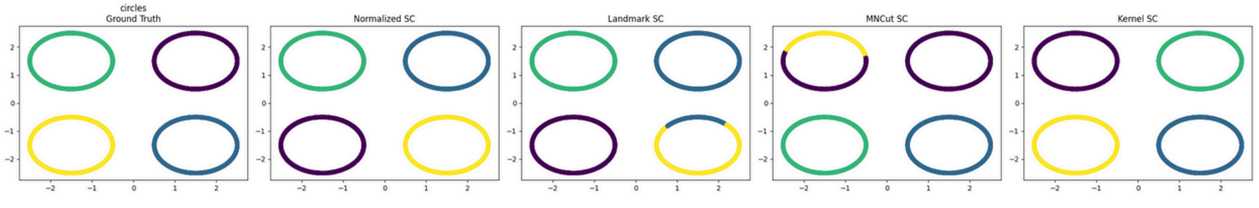
wut		"circles", "cross", "graph", "mk1", "mk2", "mk3", "trapped_lovers","twosplashes", "smile" authored by the students of Faculty of Mathematics and Information Science, Warsaw University of Technology
sipu		"a1", "a2", "aggregation", "spiral" or compiled by P. Fränti and his colleagues and research students from the University of Eastern Finland.
uni		"ecoli", "glass", "ionosphere", "sonar"," wdbc", "wine", "yeast" A selection of 8 high-dimensional datasets available at the UCI (University of California, Irvine)

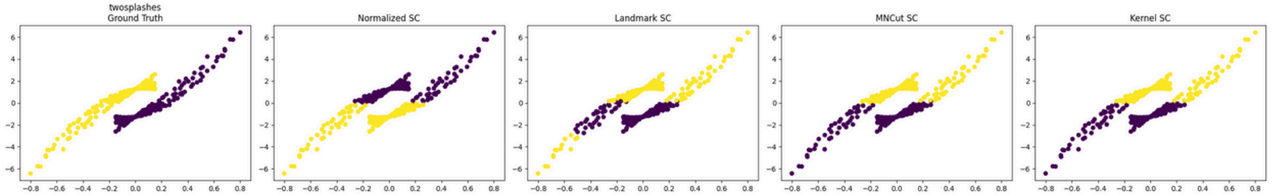
Clustering results - WUT

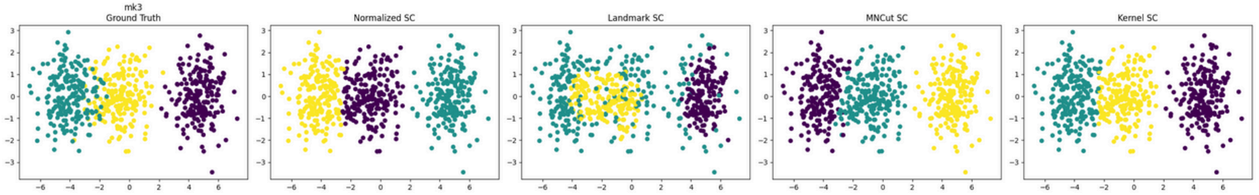
"circles", "cross", "graph", "mk1", "mk2", "mk3", "trapped_lovers","twosplashes", "smile"

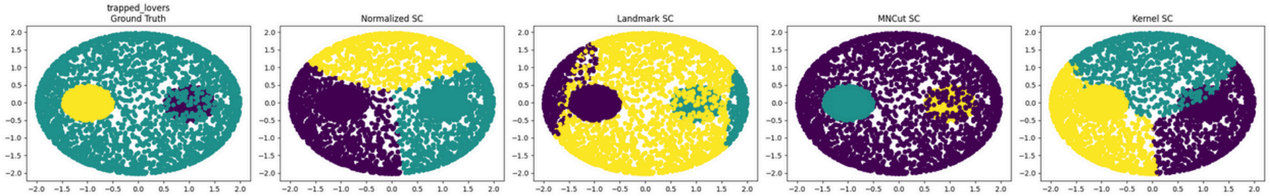


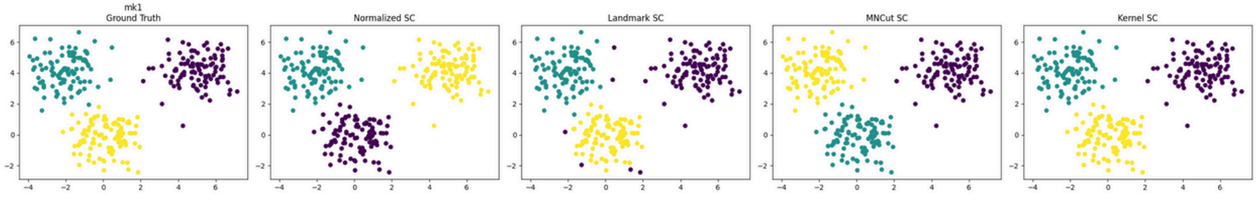


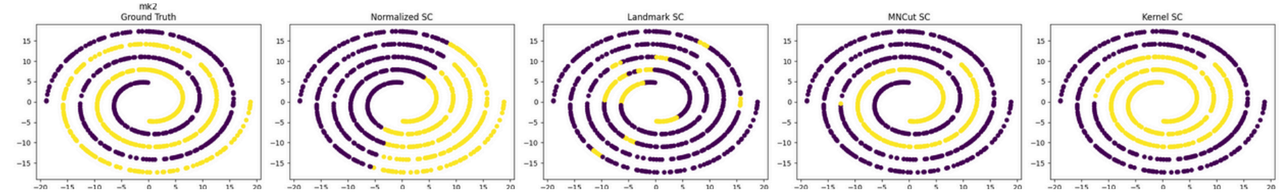


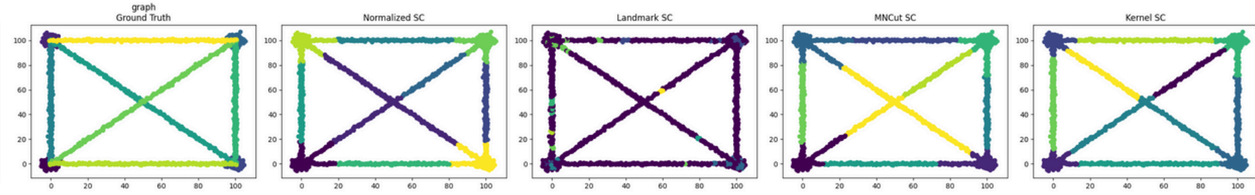






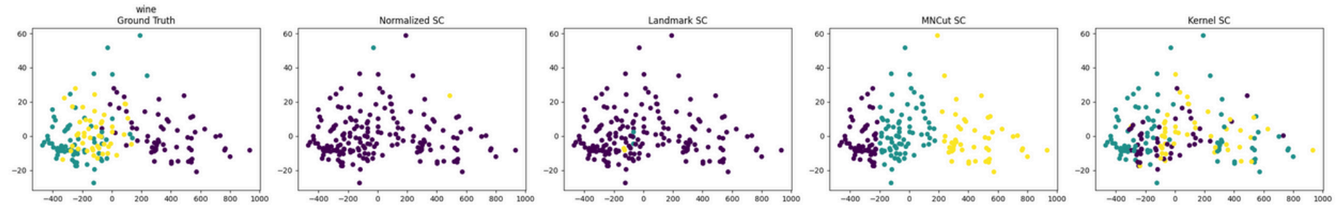


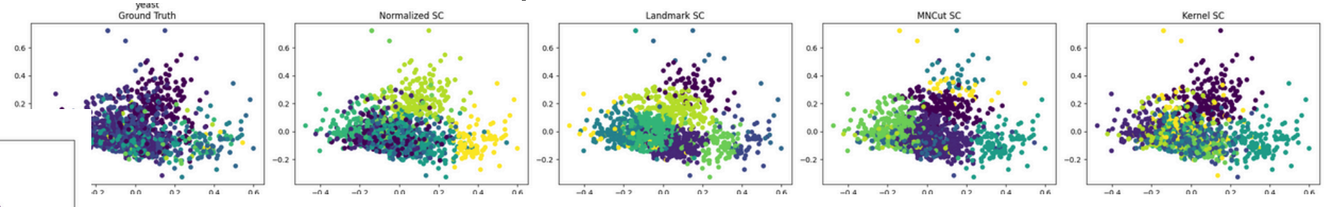


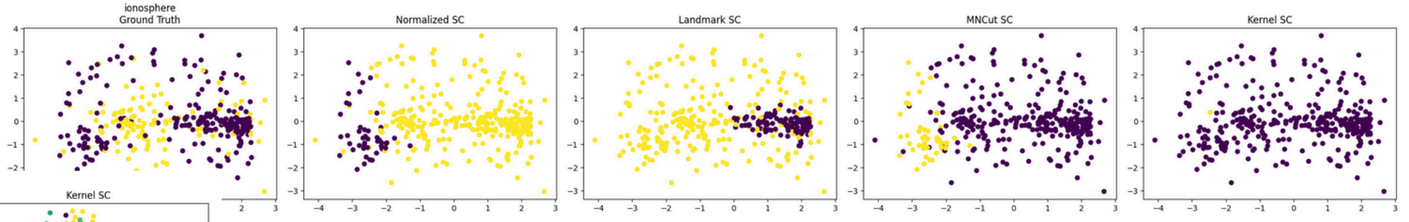


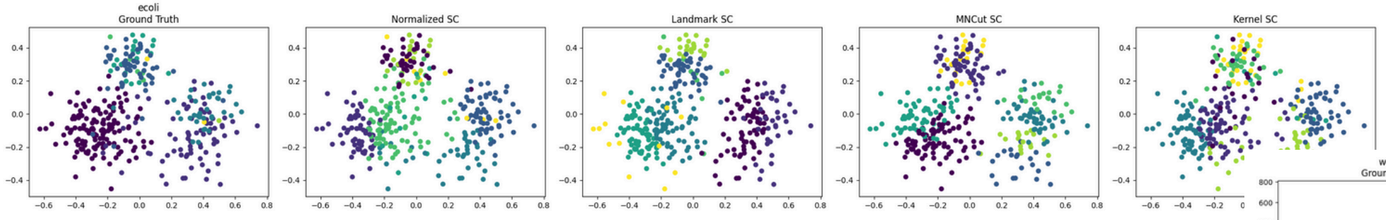
Clustering results -UCI

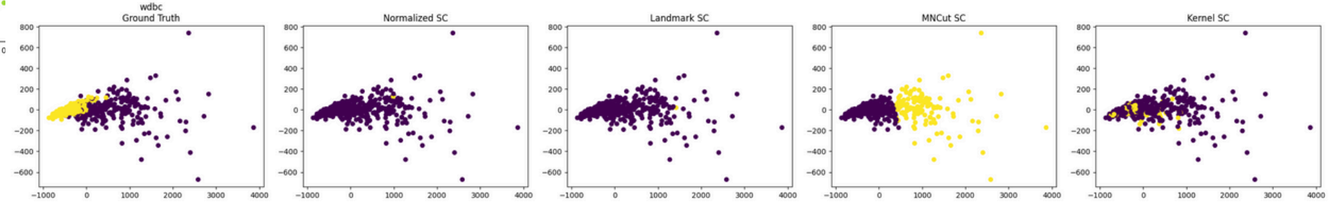
"ecoli", "glass", "ionosphere", "sonar"," wdbc", "wine", "yeast"

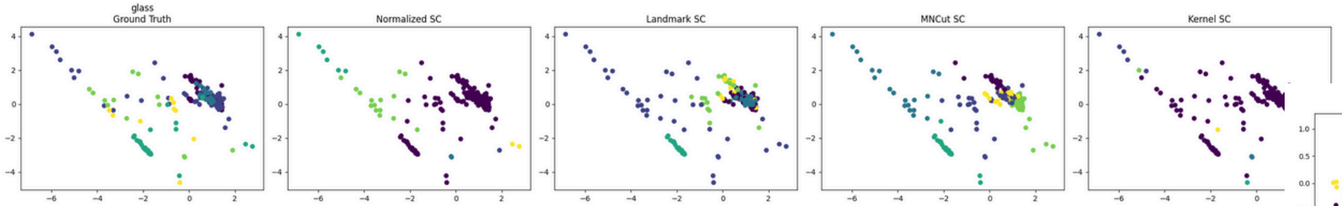


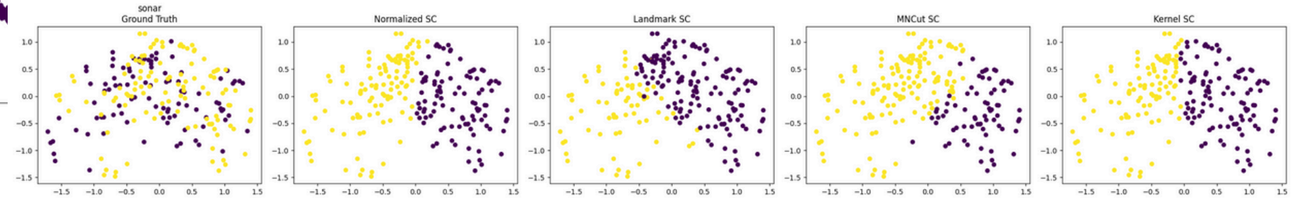






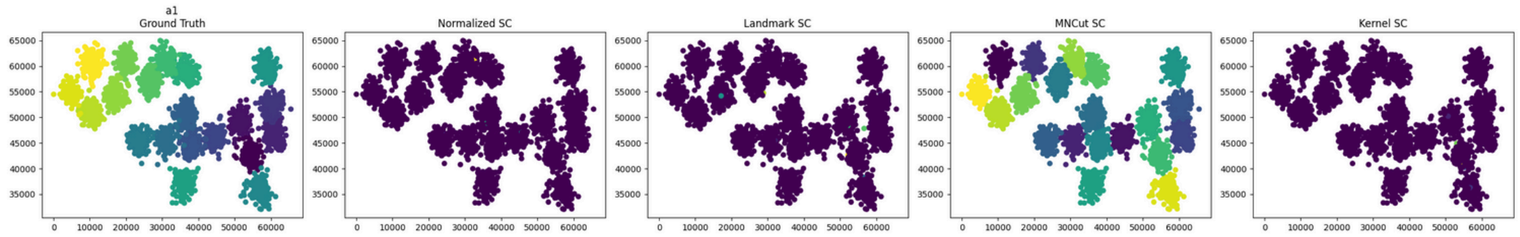


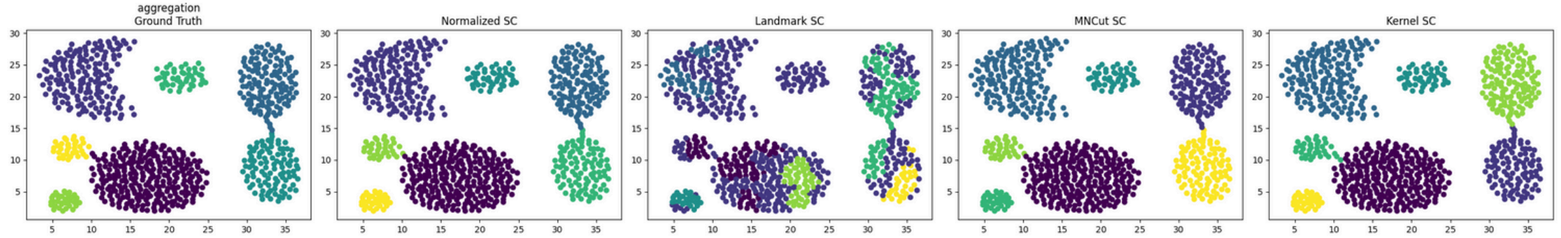


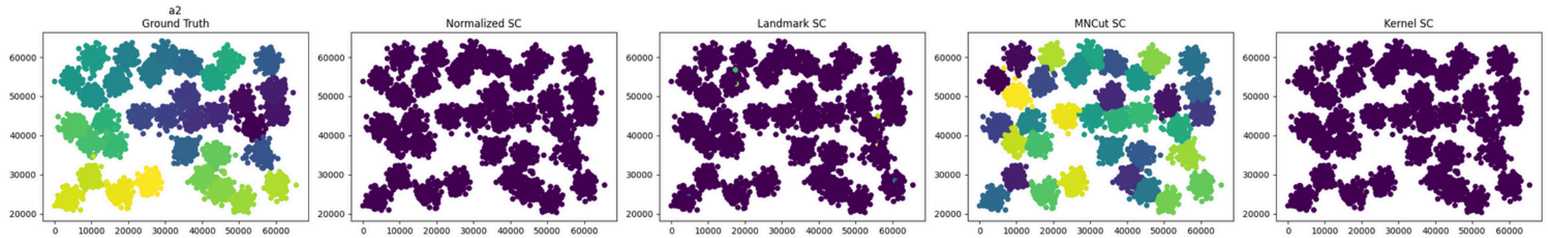


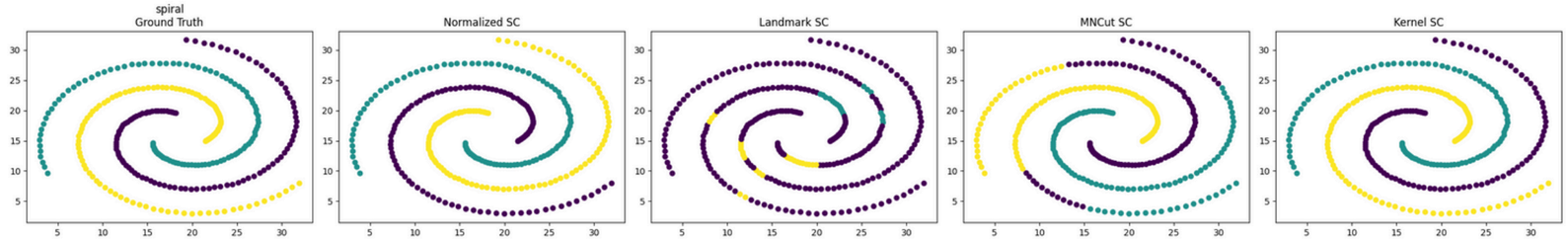
Clustering results - SIPU

a1", "a2", "aggregation", "spiral"

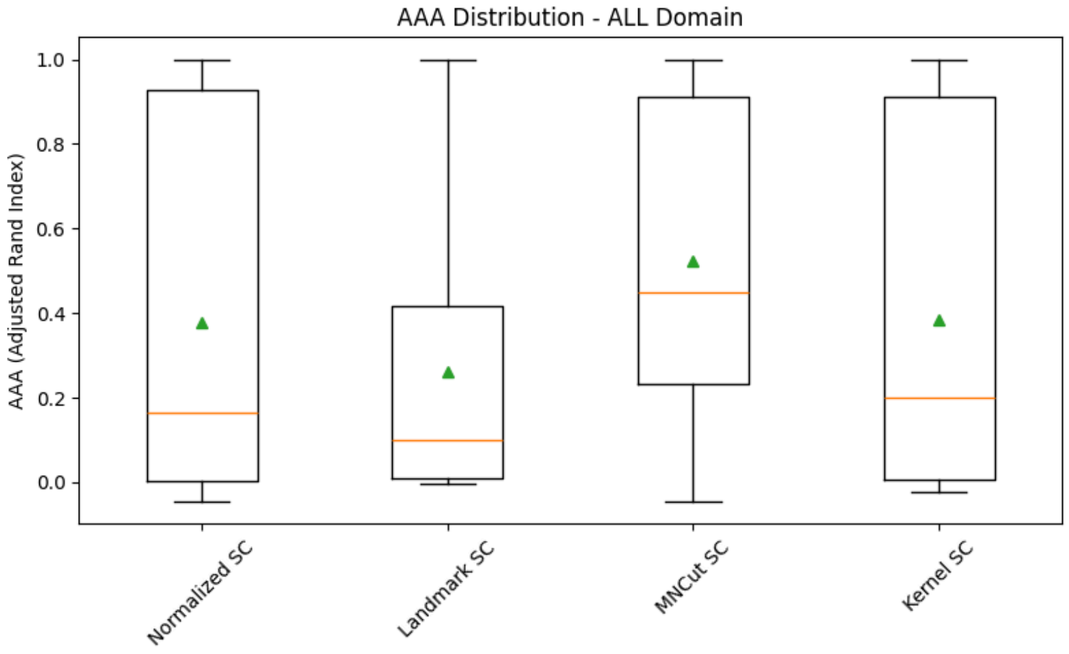








Results - ALL



Normalized Spectral Clustering:

- Has the highest **median (~0.3),** and a wide distribution, indicating relatively **high variability but with high potential clustering quality**.
- High **maximum AAA values (close to 1)**, suggesting it can achieve **near-perfect** clustering for certain datasets.
- **Means** are moderately high **(~0.45**), indicating generally **better average performance**.

Landmark SC:

- Has the lowest **median (~0.15)**, indicating generally poorer clustering quality.
- Has a **narrower distribution**, indicating consistent but generally lower performance.

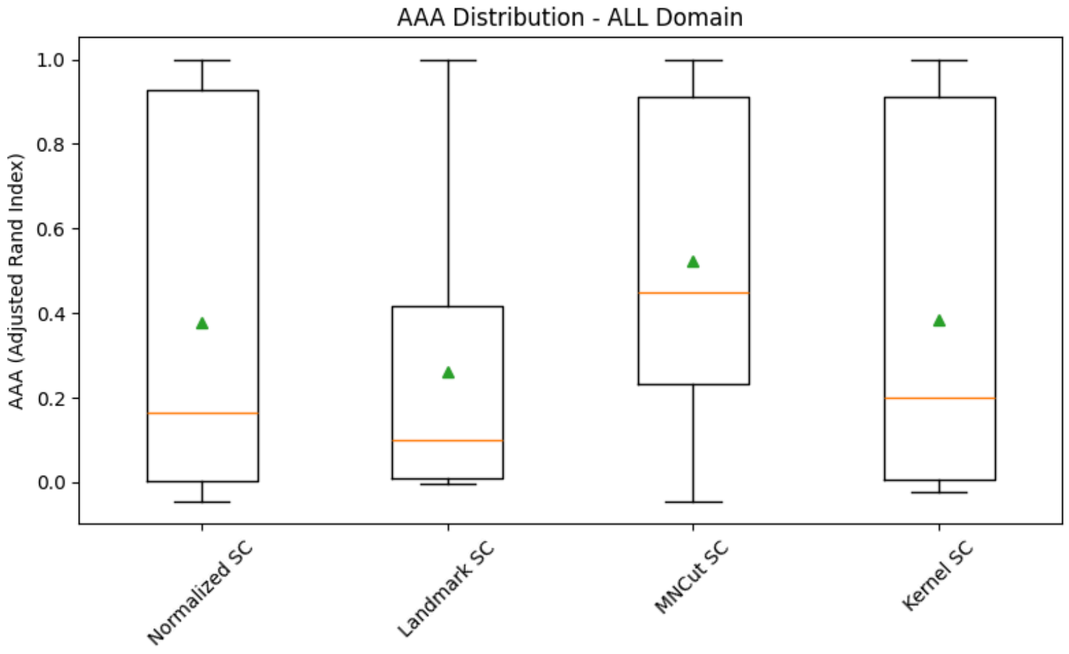
MNCut SC:

- Shows moderate **median performance (~0.4)** with relatively high consistency and moderate distribution range.
- Moderate **mean performance (~0.5)**, indicating overall good clustering quality.

Kernel SC:

- Shows a **low median (~0.2)**, but with a **higher maximum,** showing capability for **good clustering performance** in some cases.
- Wide distribution, suggesting **high variability** across datasets.

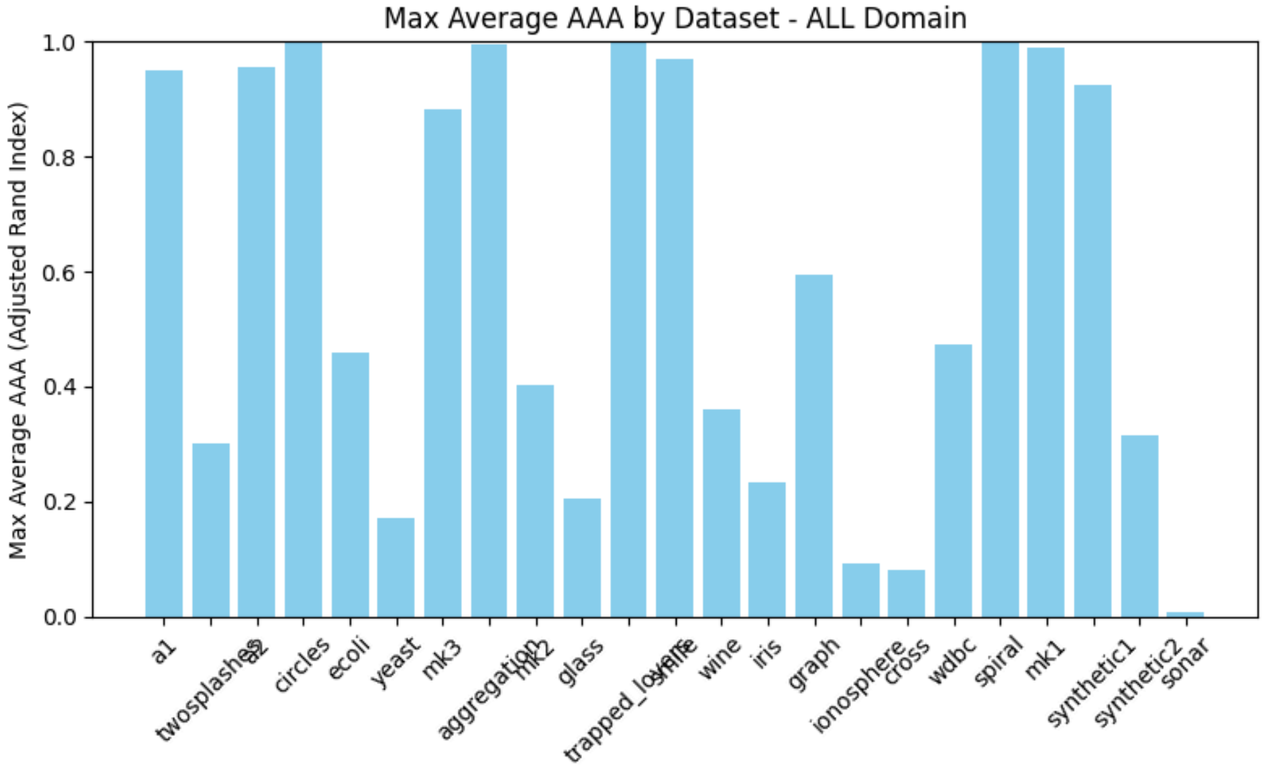
Results - ALL



Best Methods
Normalized SC and MNCut SC show generally higher clustering performance across datasets.

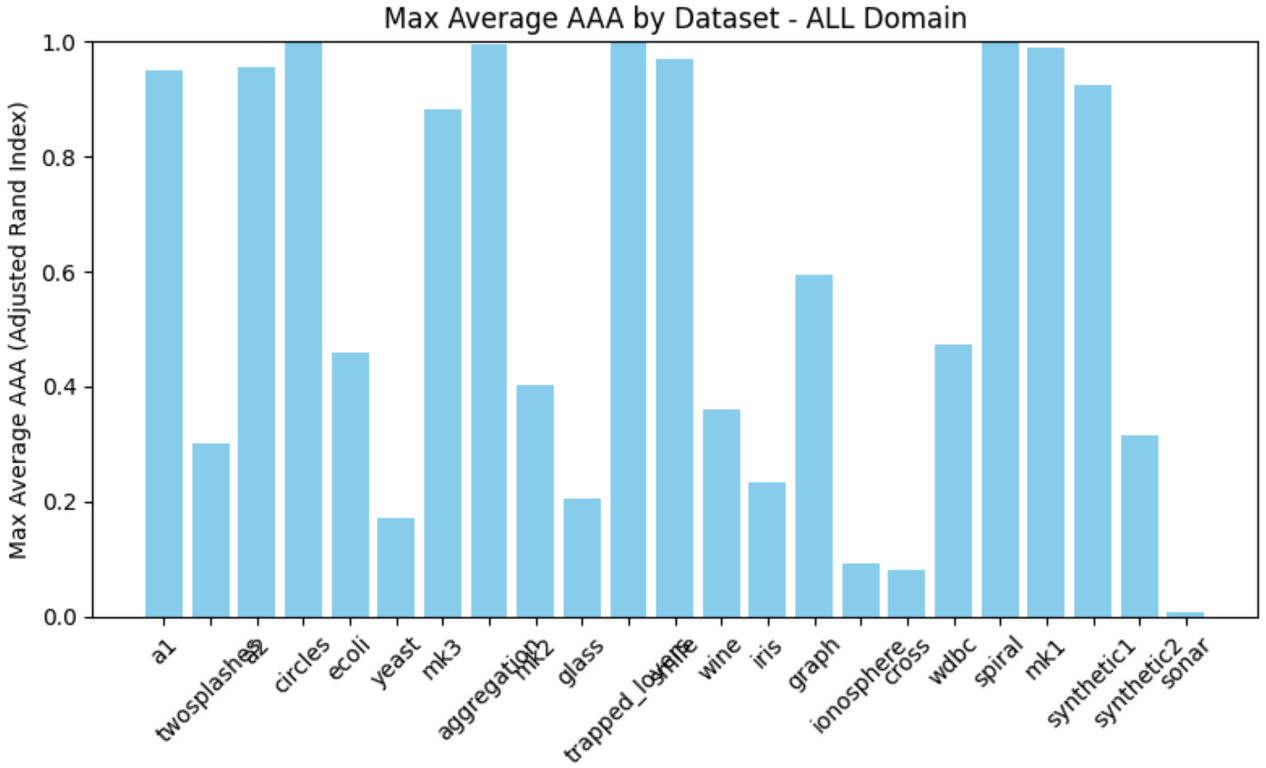
Least Effective Method:
Landmark SC generally underperforms compared to other methods.

Results – ALL



- Datasets with **High AAA (>0.8) – very good clustering quality**
  - a1, twospashes, circles, ecoli, aggregation, flame, spiral, r15, mk1
- Datasets with **Medium AAA (~0.4 to 0.7) – moderate clustering quality**
  - graph, glass, iris, synthetic1
- Datasets with **Low AAA (<0.4) – lower clustering quality**
  - yeast, wine, ionosphere, wbcd, synthetic2
- Datasets with **Very Low AAA (~0) – extremely poorly with essentially no meaningful clustering**
  - sonar

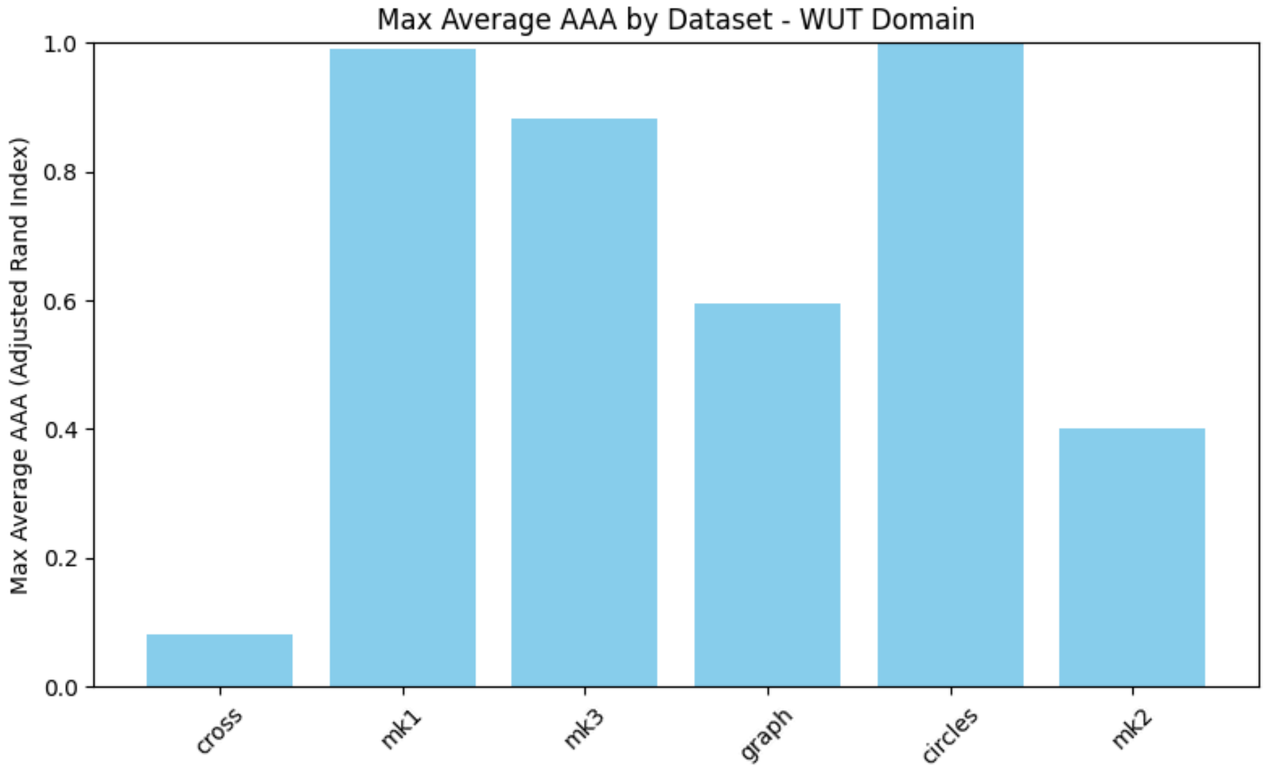
Results – ALL



Summary Based on Dataset Categories:

- **UNI** datasets: Predominantly lower clustering quality, indicating a possible difficulty in clearly separating clusters in these datasets.
- **SIPS** datasets: High clustering quality, likely due to clearly defined clusters in data structures.
- **WUT** datasets: High variability; performance strongly dataset-dependent.

Results - WUT



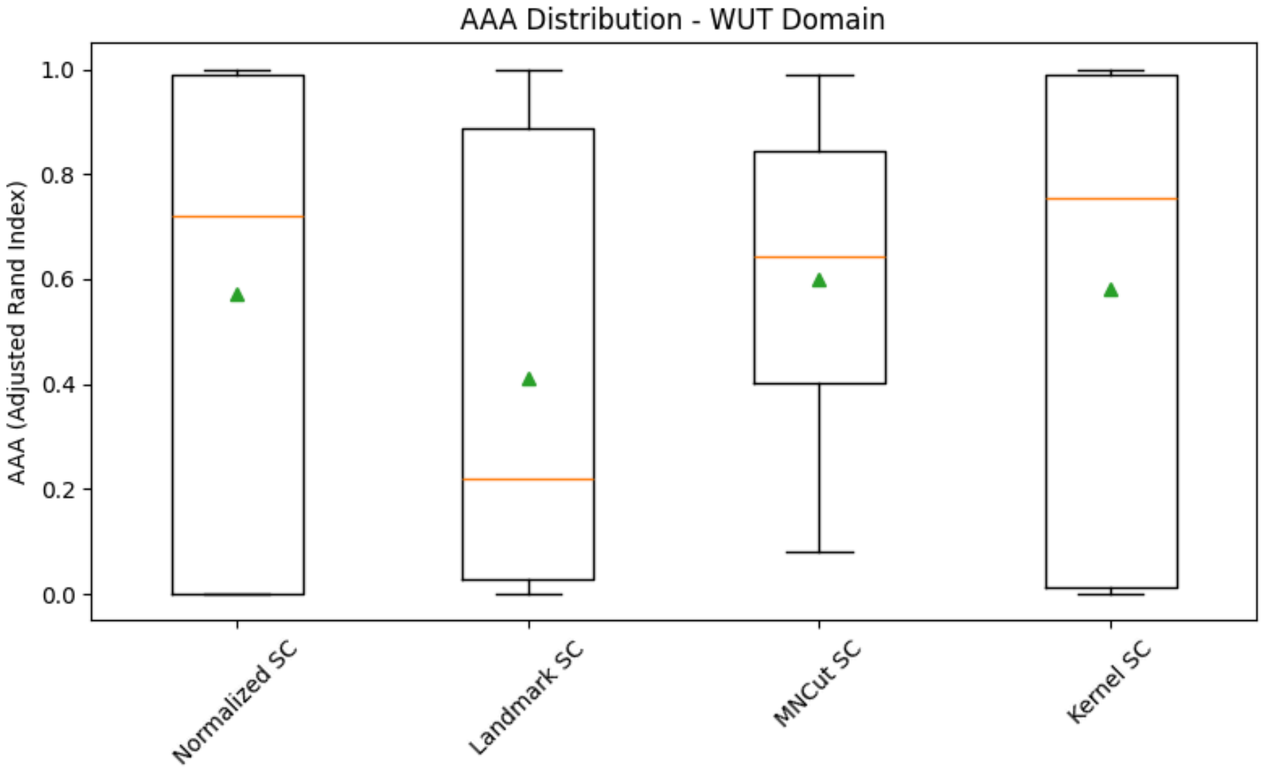
Observations

High-performance datasets:
- mk1, mk3, circles: Excellent clustering (AAA close to 1).

Medium-performance:
- graph, mk2: Good to moderate clustering.

Low-performance:
- cross: Poor clustering (AAA near zero).

Results - WUT



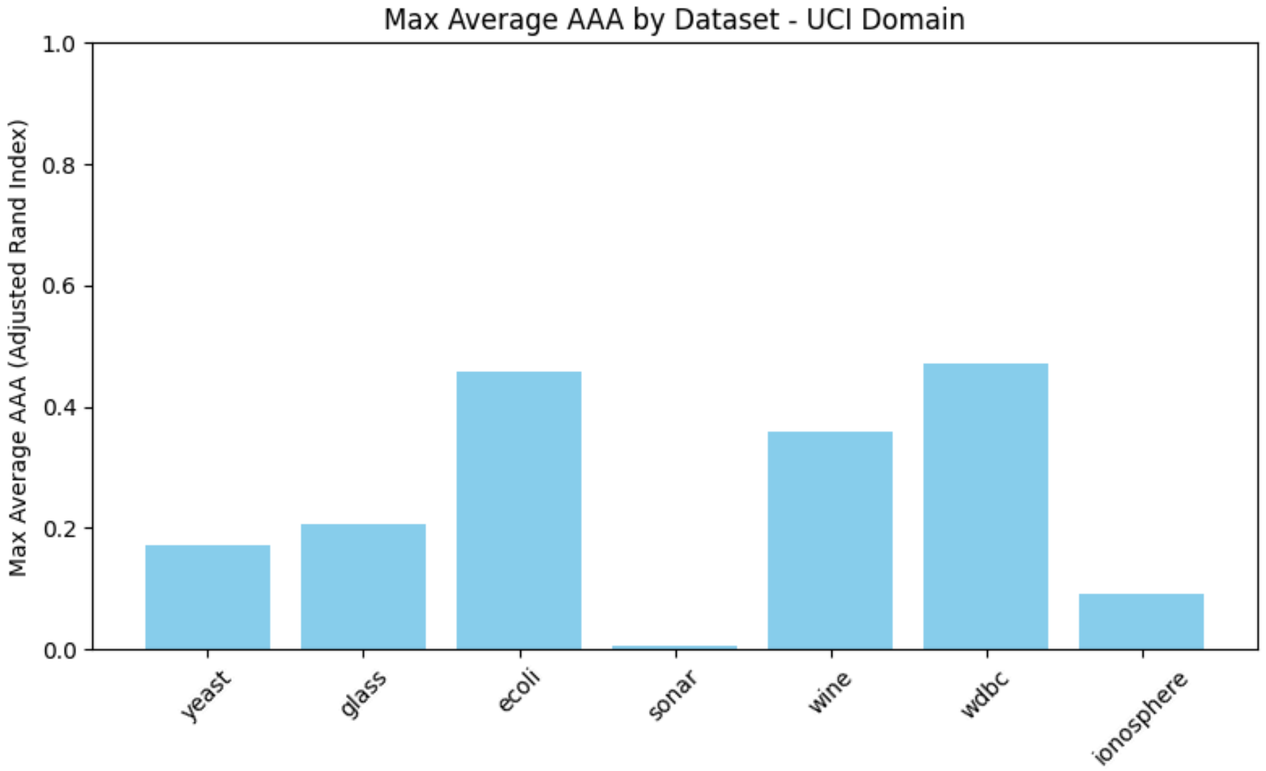
Methods Comparison

- All methods show substantial variability.
- Normalized SC and Kernel SC methods have the highest median and maximum scores, showing good reliability on WUT datasets.
- Landmark SC also performs reasonably well, although slightly lower.
- MNCut SC has a slightly narrower distribution and moderate median performance.

Conclusion for WUT Domain

Normalized SC and Kernel SC methods are most robust and generally provide superior clustering.

Results - UCI



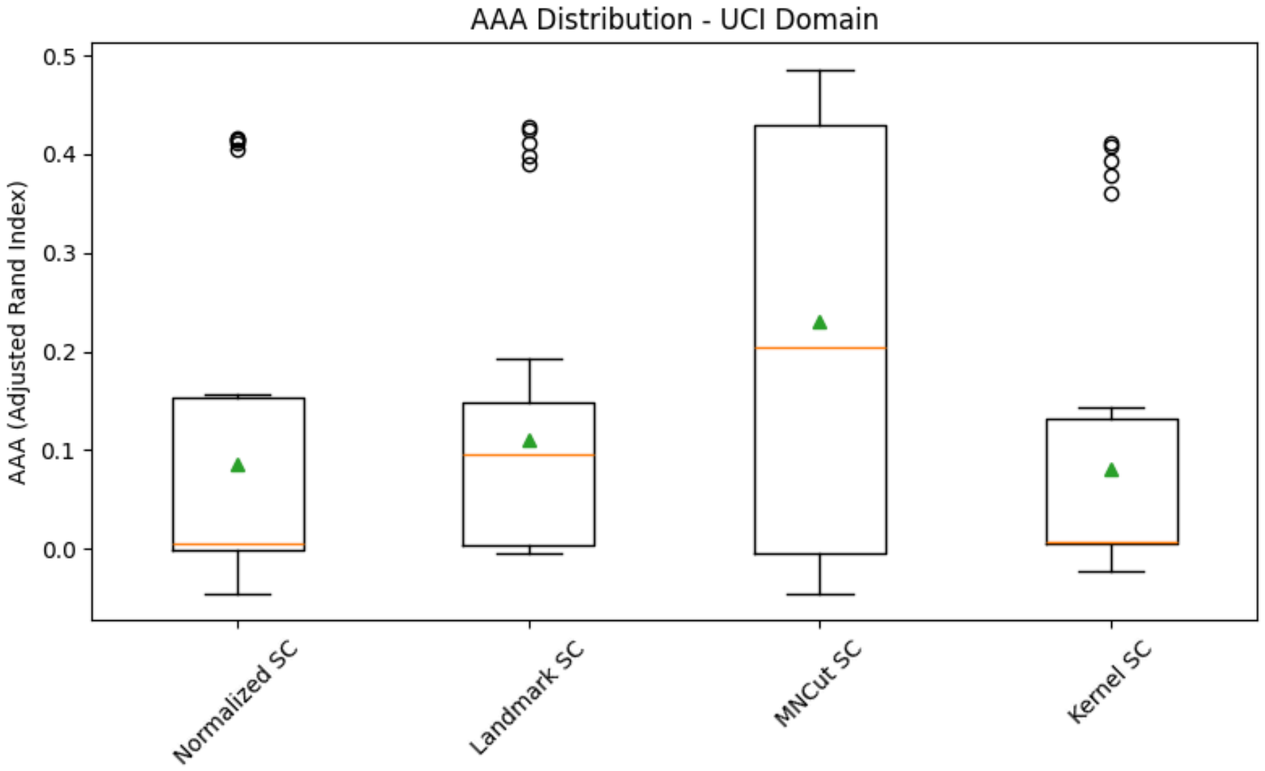
Observations

High-performance datasets:
- ecoli (~0.5 ARI), wdbc (~0.4 ARI): Best clustering

Medium-performance:
- wine, glass: Good to moderate clustering.

Low-performance:
- yeast, ionosphere, especially sonar: Poor clustering (AAA near zero).

Results - UCI



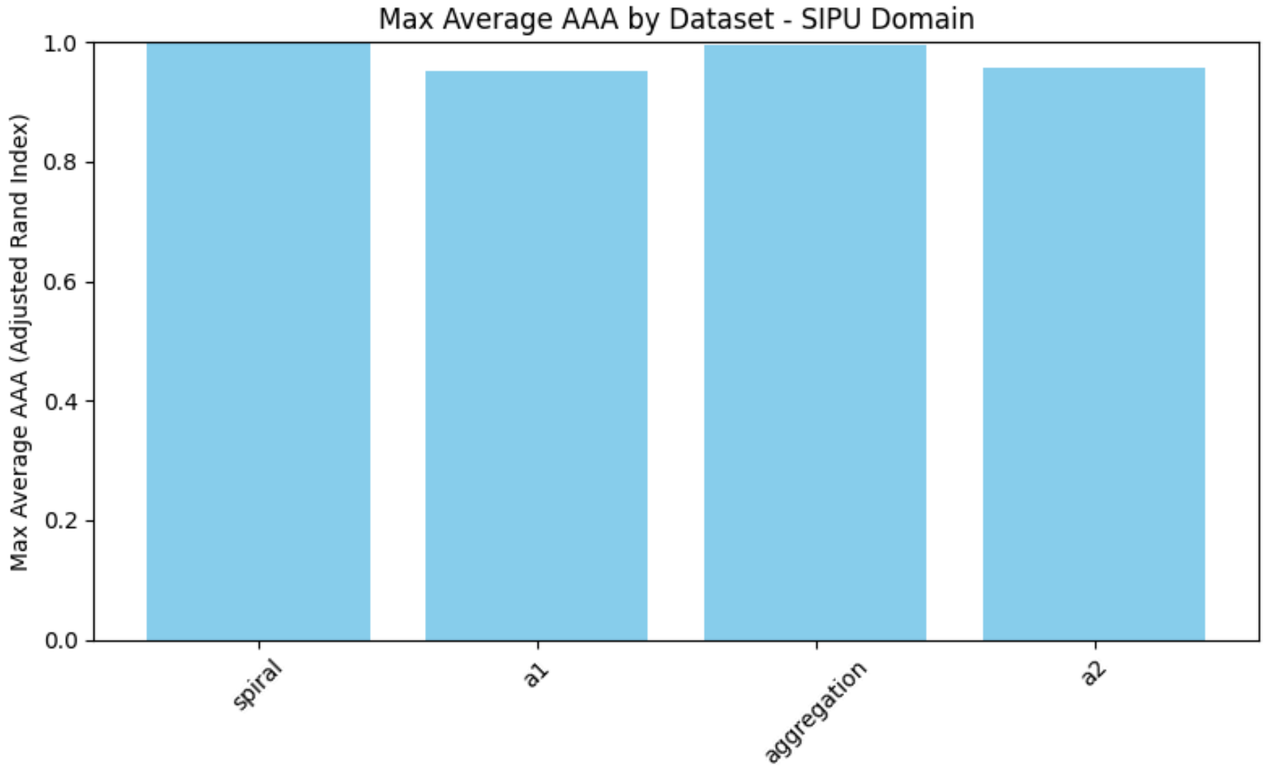
Methods Comparison

- MNCut SC clearly outperforms other methods on these datasets (higher median and mean AAA).
- Other methods (Normalized SC, Landmark SC, Kernel SC) have lower medians and considerable outliers, demonstrating poor to moderate clustering quality.

Conclusion for UCI Domain

UCI datasets are challenging for spectral clustering methods, but MNCut SC appears to provide the best performance within this categor

Results – SIPU

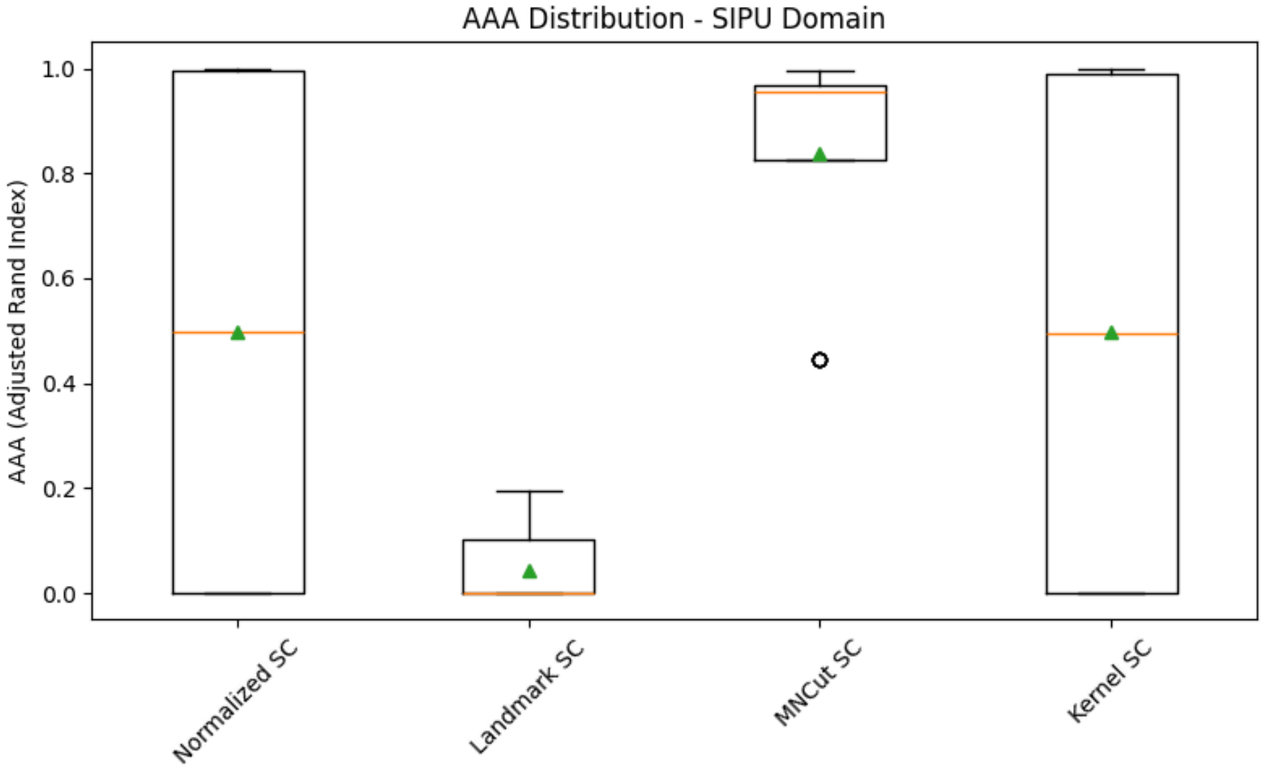


Observations

High-performance datasets:

- All provided SIPU datasets achieve near-perfect clustering (AAA very close to 1).

Results – SIPU



Methods Comparison

- MNCut SC significantly outperforms all other methods, with near-perfect median performance and minimal variability.
- Normalized SC and Kernel SC methods have wider distribution ranges but generally high maximum performance.
- Landmark SC significantly underperforms compared to others.

Conclusion for UCI Domain

SIPU datasets appear easier to cluster effectively, with MNCut SC clearly being the superior method here, achieving consistently high-quality clustering.

Conclusion

MNCut SC is particularly effective on structured datasets (especially UCI and SIPU datasets).

Normalized SC and Kernel SC perform robustly, especially on varied and moderately complex datasets (WUT domain).

Landmark SC often demonstrates relatively lower effectiveness but may still be suitable in specific, less computationally demanding scenarios.

Domain Recommended Clustering method

UCI – MNCut SC
WUT – Normalized SC, Kernel SC
SIPU – MNCut SC

Literature

- U. von Luxburg, A tutorial on spectral clustering. Stat Comput 17, 395–416 (2007)
- S. T. Wierzchoń and M. A. Kłopotek, Modern Algorithms of Cluster Analysis, Studies in Big Data 34, (chapter 5)
- T. Hastie, R. Tibshirani, J. Friedman. The elements of statistical learning: Data mining, inference, and
prediction. Springer, 2009 (chapter. 14.3.5)

Thank you for your attention!

Q&A