Spectral Bridges

Scalable Spectral Clustering free from hyperparameters

Félix Laplante   Université de Paris Saclay

Christophe Ambroise [1]   Laboratoire de Mathématiques et Modélisation d'Evry, Université Paris-Saclay, CNRS, Univ Evry,

Abstract

In this paper, Spectral-Bridges, a novel clustering algorithm, is presented. This algorithm builds upon the traditional k-means and spectral clustering frameworks by subdividing data into small Voronoi regions, which are subsequently assessed for their connectivity. Drawing inspiration from Support vector machine, a non-parametric clustering approach is proposed. This approach is characterized by minimal hyperparameters and intuitive usability, thereby augmenting adaptability and enabling the delineation of intricate, non-convex cluster structures.

Both global and local data arrangements are aimed to be discerned by Spectral-Bridges in a scale-invariant manner.

The empirical results underscore Spectral-Bridges as a fast, robust, and versatile tool for sophisticated clustering tasks spanning diverse domains. Its efficacy is observed to extend seamlessly to large-scale scenarios encompassing both real-world and synthetic datasets.

Keywords: spectral clustering, vector quantization, scalable, non-parametric

Contents

---

[1]Corresponding author: christophe.ambroise@univ-evry.fr

1 Introduction

Clustering is a fundamental technique for exploratory data analysis. It partition a set of objects into a certain number of homogeneous groups, each referred to as a cluster. It is extensively utilized across diverse fields such biology, social sciences, and psychology. Clustering is frequently employed in conjunction with supervised learning as a pre-processing step, where it helps to structure and simplify data, thereby enhancing the performance and interpretability of subsequent predictive models.

There are numerous approaches to clustering, each defined by how similarity between objects is measured, either through a similarity measure or more strictly through a distance metric.

Density-based methods identify regions within the data that have a high concentration of points, corresponding to the modes of the joint density. A notable non-parametric example of this approach is DBSCAN (Ester et al. 1996). In contrast, model-based clustering, such as Gaussian mixture models, represents a parametric approach to density-based methods.

Geometric approaches, such as kmeans (MacQueen et al. 1967) are distance-based and aim to partition the data in a way that optimizes a criterion reflecting group homogeneity.

Graph-based methods treat data as a graph, with vertices representing data points and edges weighted to reflect the affinity between these points.

The algorithm proposed in this paper draws from numerous clustering techniques. The initial intuition is to detect high-density areas. To this end, a vector quantization is used to divide the space into a Voronoi tessellation. An original geometric criterion is then employed to detect pairs of Voronoi regions that are either distant from each other or separated by a low-density boundary. Finally, this affinity measure is considered as the weight of an edge in a complete graph connecting the centroids of the tessellation, and a spectral clustering algorithm is used to find a partition of this graph.

The paper begins with a section dedicated to presenting the context and related algorithms, followed by a detailed description of the proposed algorithm. Experiments and comparisons with reference algorithms are then conducted on both real and synthetic data.

2 Background

Spectral clustering is a graph-based approach that computes the eigenvectors of the graph's Laplacian matrix. This technique transforms the data into a lower-dimensional space, making the clusters more discernible. A standard algorithm like k-means is then applied to these transformed features to identify the clusters(Von Luxburg 2007).

It enables to capture complex data structures and discern clusters based on the connectivity of data points in a transformed space. Notice that spectral clustering can be seen as a relaxed graph cut problem.

3 Spectral bridges

The proposed algorithm uses K-means centroids for vector quantization defining Voronoi region, and a strategy is proposed to link these regions, with the "affinity" gauged in terms of minimal margin between pairs of classes. These affinities are considered as weight of edges defining a completely connected graph whose vertices are the regions. Spectral clustering on the region provide a partition of the input space. The sole parameters of the algorithm are the number of Voronoi region and the number of final cluster.

3.1 Bridge gain affinity

The basic idea involves calculating the difference in inertia achieved by projecting onto a segment connecting two centroids, rather than using the two centroids separately. If the difference is small, it suggests a low density between the classes. Conversely, if this diffrence is large, it indicates that the two classes may reside within the same densely populated region.
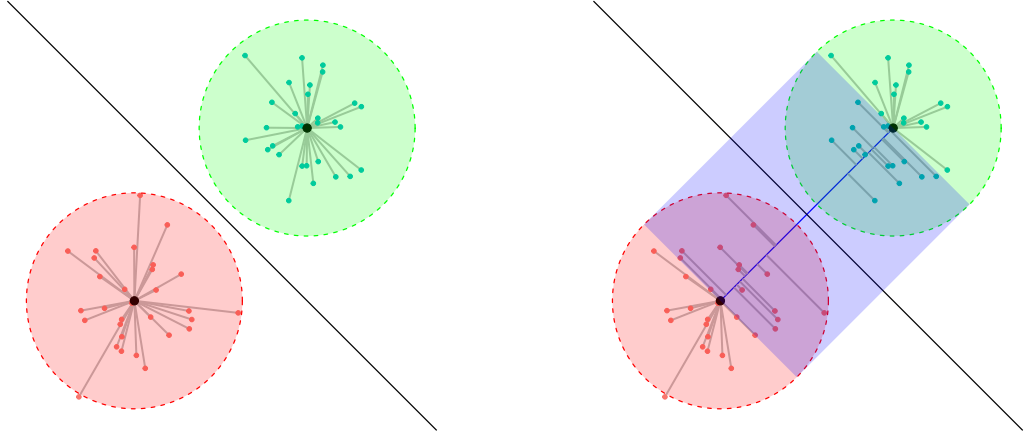

Figure 1: Balls (left) versus Bridge (right). The inertia of each structure is the sum of the squared distances represented by grey lines.

The inertia of two balls k and l is

$$I_{kl} = \sum_{i \in k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 + \sum_{i \in l} \|\boldsymbol{x}_i - \boldsymbol{\mu}_l\|^2.$$

The inertia of a bridge between k and l is defined as

$$B_{kl} = \sum_{i \in kl} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2,$$

where

$$\boldsymbol{p}_{kl}(\boldsymbol{x}_i) = \boldsymbol{\mu}_k + t_i(\boldsymbol{\mu}_l - \boldsymbol{\mu}_k),$$

with

$$t_i = \min\left(1, \max\left(0, \frac{\langle \boldsymbol{x}_i - \boldsymbol{\mu}_k | \boldsymbol{\mu}_l - \boldsymbol{\mu}_k \rangle}{\|\boldsymbol{\mu}_l - \boldsymbol{\mu}_k\|^2}\right)\right).$$

The normalized average of the difference betwenn Bridge and balls inertia is (See [Appendix](#))

$$\frac{B_{kl} - I_{kl}}{(n_k + n_l)\|\mu_k - \mu_l\|^2} = \frac{\sum_{i \in k}\langle x_i - \mu_k | \mu_l - \mu_k\rangle_+^2 \sum_{i \in l}\langle x_i - \mu_l | \mu_k - \mu_l\rangle_+^2}{(n_k + n_l)\|\mu_k - \mu_l\|^4}.$$

From this reduction, we define the bridge affinity between centroids k and l as:

$$a_{kl} = \begin{cases} 0, & \text{if } k = l, \\ \sqrt{\frac{B_{kl} - I_{kl}}{(n_k + n_l)\|\mu_k - \mu_l\|^2}}, & \text{otherwise.} \end{cases}$$

The basic intuition behind this affinity is that t_i represents the relative position of the projection of x_i on the segment $[\mu_k, \mu_l]$. For each x_i an affinity value α_i is defined as

$$\alpha_i = \begin{cases} t_i, & \text{if } t_i \in [0, 1/2] \\ 1 - t_i, & \text{if } t_i \in]1/2, 1], \end{cases}$$

This value represents the relative position on the segment, with the centroid of the class to which x_i belongs as the starting point.

The boundary that separates the two clusters defined by centroids μ_k and μ_l is a hyperplane. This hyperplane is orthogonal to the line segment connecting the centroids and intersects this segment at its midpoint.

If we consider all points $x_i \in kl$ which are not projected on centroids but somewhere on the segments, The distance from a point to the hyperplane is large,

$$\|p_{kl}(x_i) - \mu_{kl}\| = (1/2 - \alpha_i)\|\mu_k - \mu_l\|.$$

This distance is similar to the concept of margin in Support Vector Machine (Cortes and Vapnik 1995).

When the α_i values are small (close to zero since $\alpha_i \in [0, 1/2]$), the margins to the hyperplane are large, indicating a low density between the classes. Conversely, if the margins are small, it suggests that the two classes may reside within the same densely populated region. Consequently, the sum of the α_i or α_i^2 increases with the density of the region between the classes.

Note that the criterion is local and indicates the relative difference in densities between the balls and the bridge, rather than evaluating a global score for the densities of the structures.

3.2 Algorithm

4 Numerical experiments

In this section, we present the results obtained from testing our algorithm on various datasets, both small and large scale, including real-world and well-known synthetic datasets. These experiments assess the accuracy, time and space complexity, ease of use, robustness, and adaptability of our algorithm. We compare **Spectral-Bridges (SB)** against several state-of-the-art methods, including **k-means++ (KM)** (MacQueen et al. 1967; **arthur2007k?**), **Expectation-Maximization (EM)** (**dempster1977maximum?**), **Ward Clustering (WC)** (**ward1963hierarchical?**), and **DBSCAN (DB)** (Ester et al. 1996). This comparison establishes baselines across centroid-based clustering algorithms, hierarchical methods, and density-based methods. We evaluate the algorithms on both raw and PCA-processed data with varying dimensionality. For synthetic datasets, we introduce Gaussian and/or uniform noise to evaluate the robustness of our algorithm.

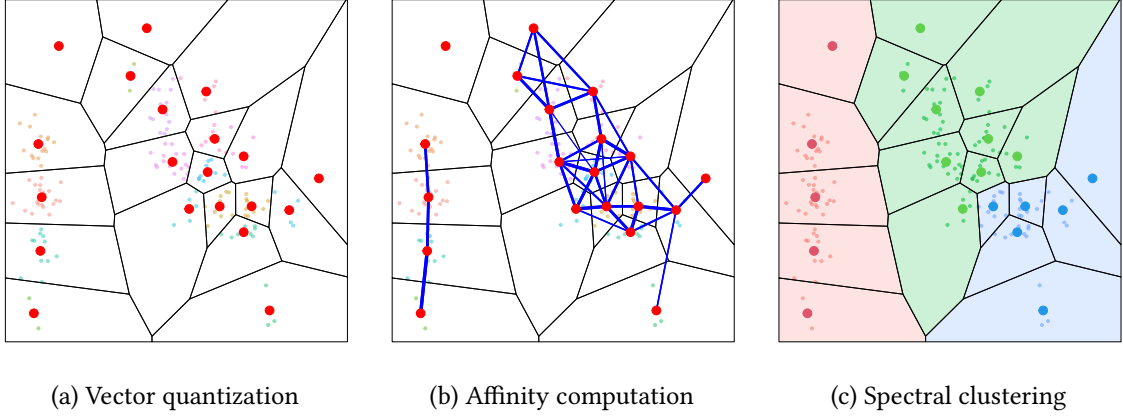|		
(a) Vector quantization	(b) Affinity computation	(c) Spectral clustering

Figure 2: Illustration of the Spectral bridges algorithm with the Iris dataset (first principal plane). Vector quantization (left), Affinity computation (center), Spectral clustering and spreading (right).

Algorithm 1 Spectral Bridges

1: **procedure** SpectralBridges(X, k, m) ▷ X: input dataset, k: number of clusters, m: number of Voronoi regions
2:     **Step 1: Vector Quantization**
3:     $centroids \leftarrow$ KMeans(X, m) ▷ Initial centroids using k-means++
4:     $voronoiRegions \leftarrow$ Subdivide($X, centroids$) ▷ Subdivide data into Voronoi regions
5:     **Step 2: Affinity Matrix Computation** $A = \{a_{kl}\}$
6:     **Step 3: Spectral Clustering** ▷ Affect each region to a cluster
7:     $labels \leftarrow$ SpectralClustering(X, k)
8:     **Step 4: Propagate** ▷ Each data point is affected to the cluster of its region
9:     $clusters \leftarrow$ Propagate($X, labels, centroids$)
10:     **return** $clusters$ ▷ Cluster labels for data points in X
11: **end procedure**

4.1 Real-world Data

- **MNIST**: A large dataset containing 60,000 handwritten digit images in ten balanced classes, commonly used for image processing benchmarks. Each image consists of $28 \times 28 = 784$ pixels.
- **UCI ML Breast Cancer Wisconsin**: A dataset featuring computed attributes from digitized images of fine needle aspirates (FNA) of breast masses, used to predict whether a tumor is malignant or benign.

4.2 Synthetic Data

- **Impossible**: A synthetic dataset designed to challenge clustering algorithms with complex patterns.
- **Moons**: A two-dimensional dataset with two interleaving half circles.
- **Circles**: A synthetic dataset of points arranged in two non-linearly separable circles.
- **Smile**: A synthetic dataset with points arranged in the shape of a smiling face, used to test the separation of non-linearly separable data.

4.2.1 Datasets Summary & Class Balance

Table 1: Datasets Summary & Class Balance

Dataset	#Dims	#Samples	#Classes	Class Proportions
MNIST	784	60000	10	9.9%, 11.2%, 9.9%, 10.3%, 9.7%, 9%, 9.9%, 10.4%, 9.7%, 9.9%
Breast Cancer	30	569	2	37.3%, 62.7%
Impossible	2	3594	7	24.8%, 18.8%, 11.3%, 7.5%, 12.5%, 12.5%, 12.5%
Moons	2	1000	2	50%, 50%
Circles	2	1000	2	50%, 50%
Smile	2	1000	4	25%, 25%, 25%, 25%

Class proportions are presented in ascending order starting from label 0.

4.3 Metrics

To evaluate the performance of our clustering algorithm, we use the Adjusted Rand Index (**ARI**) (**halkidi2002cluster?**) and Normalized Mutual Information (**NMI**) (**cover1991information?**). ARI measures the similarity between two clustering results, ranging from −0.5 to 1, with 1 indicating perfect agreement. NMI ranges from 0 to 1, with higher values indicating better clustering quality. In some tests, we also report the variability of scores across multiple runs due to the random initialization in k-means, though k-means++ generally provides stable and reproducible results.

4.4 Platform

All experiments were conducted on an Archlinux machine with Linux 6.9.3 Kernel, 8GB of RAM, and an AMD Ryzen 3 7320U processor.

4.5 Accuracy

We first evaluated our algorithm's accuracy on the MNIST dataset. Metrics were collected to compare our method with k-means++, EM, and Ward clustering. Metric were estimated by taking the empirical average over 100 consecutive runs with the same random seed for each method.

Let h denote the embedding dimension of the dataset. We tested our method on the raw MNIST dataset without preprocessing ($h = 784$) and after reducing its dimension using PCA to $h \in \{8, 16, 32, 64\}$ (see fig.1).
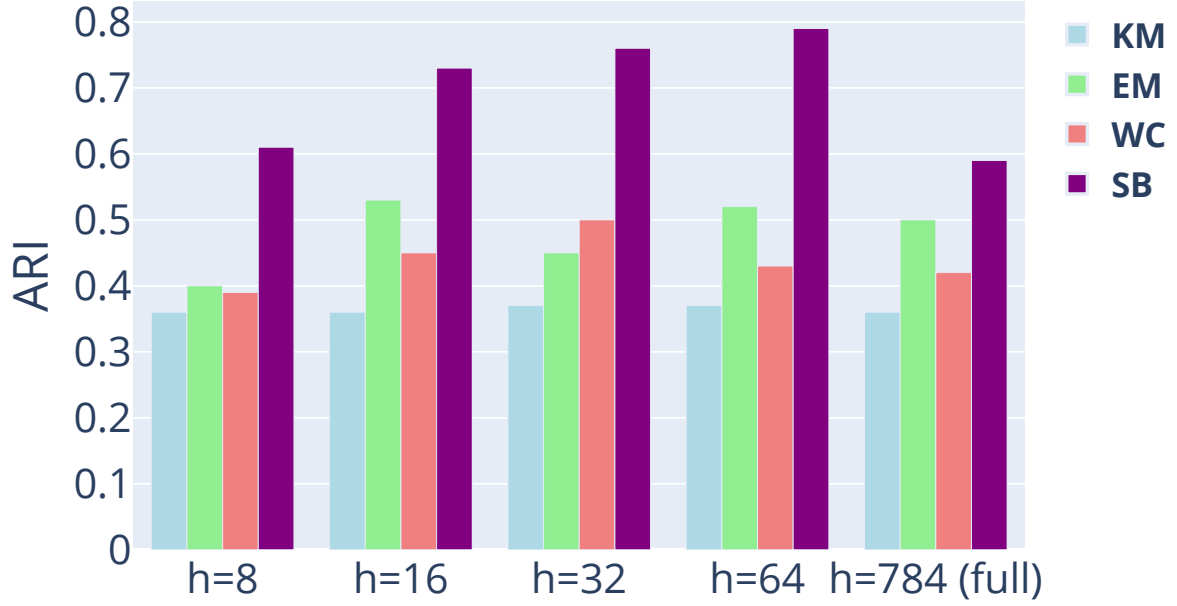
Figure 3: Comparison of **k-means++** (blue), **EM** (green), **Ward Clustering** (red), and **Spectral-Bridges** (purple) on PCA embedding and full MNIST

For visualization purposes, we projected with UMAP the predicted clusters from our algorithm and other methods to compare them against the ground truth labels to better understand the cluster shapes (see table 2). Note that the projection was not used in our experiments as an embedding, and thus does not play any role in the clustering process itself. As a matter of fact, the embedding used was obtained with PCA, $h = 64$. Note that the label colors match the legend only in the case of the ground truth data. Indeed, the ordering of the labels have no impact on clustering quality.

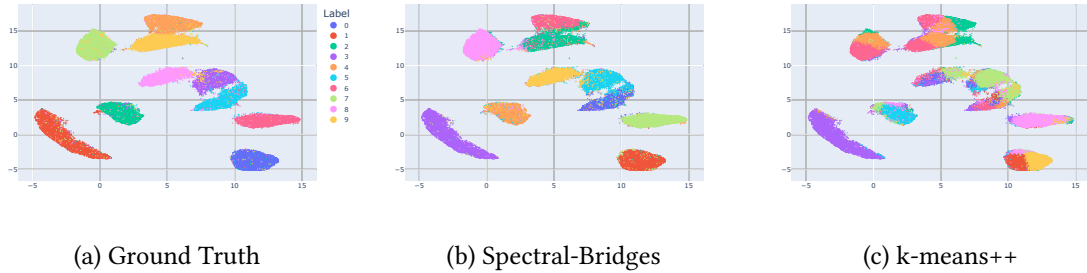(a) Ground Truth   (b) Spectral-Bridges   (c) k-means++

Figure 4: UMAP projection of predicted clusters: **Ground Truth (left)**, **Spectral-Bridges (center)**, **k-means++ (right)** UMAP projection of predicted clusters : **Ground Truth (top)**, **Spectral-Bridges (middle)**, **k-means++ (bottom)**
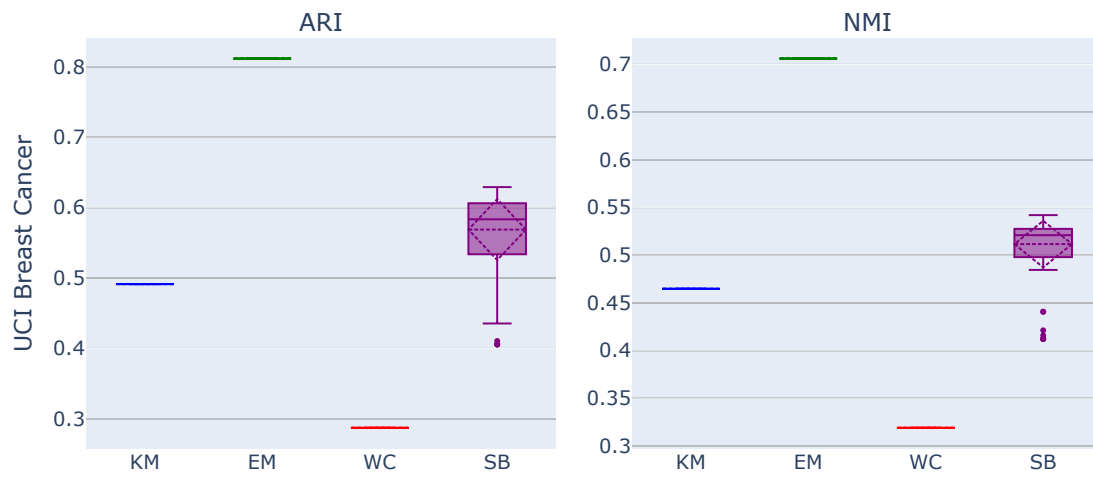
TODO CANCER

Figure 5: ARI and NMI scores of **k-means++** (blue), **EM** (green), **Ward Clustering** (red), and **Spectral-Bridges** (purple) on the UCI Breast Cancer dataset
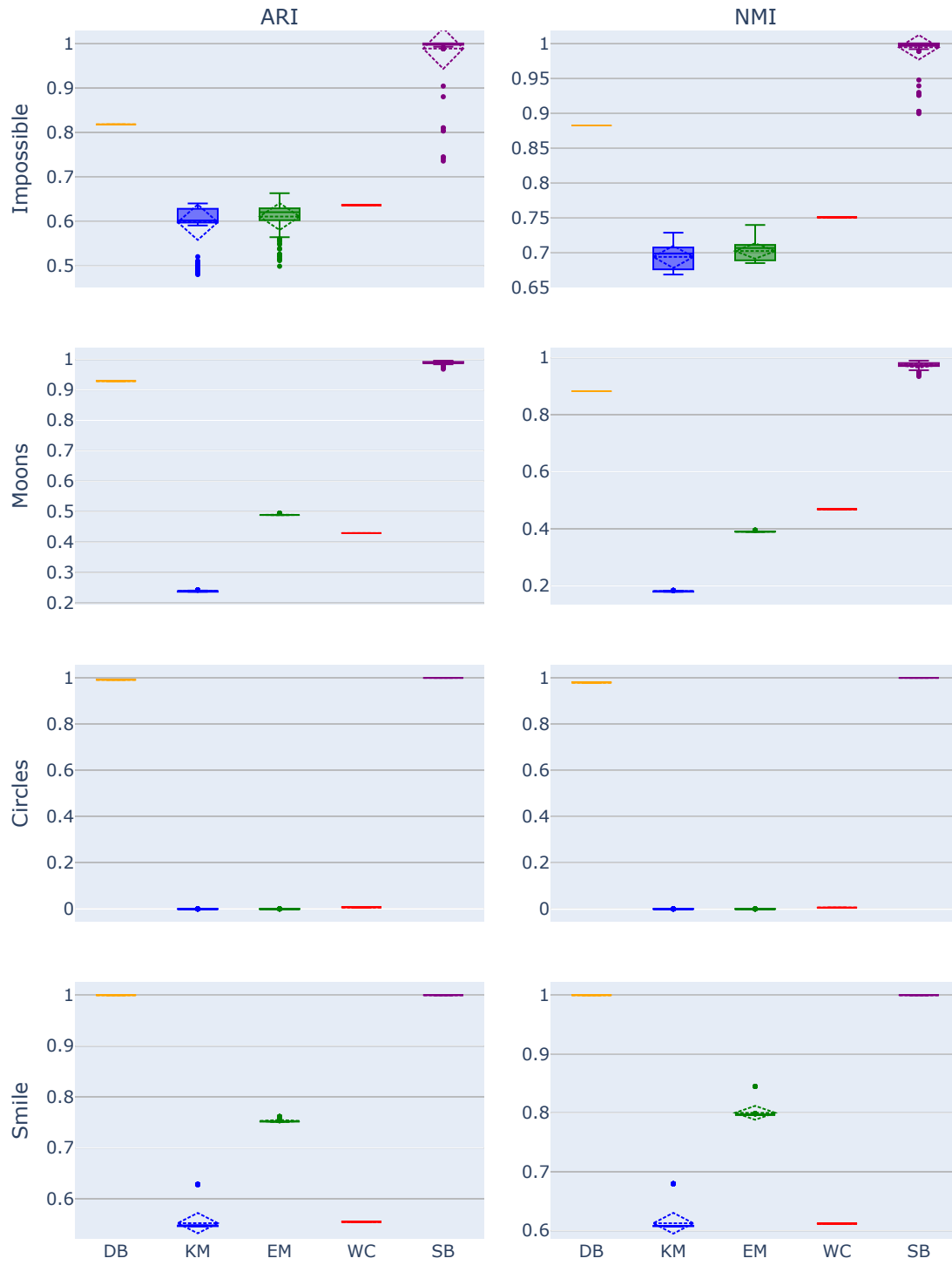
140   …..

Figure 6: TODO

5 Conclusive remarks

Possibility to kernelize

6 Appendix

6.1 Derivation of the bridge gain

Notice that B_{kl}, the bridge inertia between centroids k and l, can be expressed as the sum of three terms:

$$B_{kl} = \sum_{i|t_i=0} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 + \sum_{i|t_i=1} \|\boldsymbol{x}_i - \boldsymbol{\mu}_l\|^2 + \sum_{i|t_i\in]0,1[} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2.$$

The last term may be decomposed in two parts

$$\sum_{i|t_i\in]0,1[} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2 = \sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2 + \sum_{i|t_i\in [\frac{1}{2},1[} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2$$

and each part further decomposed using Pythagore

$$\sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2 = \sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 - \sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{\mu}_k - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2$$

$$= \sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 - \sum_{i|t_i\in]0,\frac{1}{2}[} \|t_i(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l)\|^2,$$

$$\sum_{i|t_i\in]\frac{1}{2},1[} \|\boldsymbol{x}_i - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2 = \sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{x}_i - \boldsymbol{\mu}_l\|^2 - \sum_{i|t_i\in]0,\frac{1}{2}[} \|\boldsymbol{\mu}_l - \boldsymbol{p}_{kl}(\boldsymbol{x}_i)\|^2$$

$$= \sum_{i|t_i\in]\frac{1}{2},1[} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 - \sum_{i|t_i\in]0,\frac{1}{2}[} \|(1-t_i)(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l)\|^2$$

Thus

$$B_{kl} - I_{kl} = \sum_{i|t_i\in]0,\frac{1}{2}[} t_i^2 \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2 + \sum_{i|t_i\in]\frac{1}{2},1[} (1-t_i)^2 \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2,$$

$$\frac{B_{kl} - I_{kl}}{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2} = \sum_{i|t_i\in]0,\frac{1}{2}[} t_i^2 + \sum_{i|t_i\in]\frac{1}{2},1[} (1-t_i)^2,$$

$$\frac{B_{kl} - I_{kl}}{(n_k + n_l)\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2} = \frac{\sum_{i\in k}\langle \boldsymbol{x}_i - \boldsymbol{\mu}_k|\boldsymbol{\mu}_l - \boldsymbol{\mu}_k\rangle_+^2 + \sum_{i\in l}\langle \boldsymbol{x}_i - \boldsymbol{\mu}_l|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\rangle_+^2}{(n_k + n_l)\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^4}.$$

References

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–97.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In *Kdd*, 96:226–31.

MacQueen, James et al. 1967. "Some Methods for Classification and Analysis of Multivariate Observations." In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–97. Oakland, CA, USA.

Von Luxburg, Ulrike. 2007. "A Tutorial on Spectral Clustering." *Statistics and Computing* 17: 395–416.

Session information

```
R version 4.4.1 (2024-06-14 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 22631)

Matrix products: default


locale:
[1] LC_COLLATE=French_France.utf8  LC_CTYPE=French_France.utf8
[3] LC_MONETARY=French_France.utf8 LC_NUMERIC=C
[5] LC_TIME=French_France.utf8

time zone: Europe/Paris
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.4.1   fastmap_1.2.0   cli_3.6.3       tools_4.4.1
 [5] htmltools_0.5.8.1 yaml_2.3.8      rmarkdown_2.27  knitr_1.47
 [9] jsonlite_1.8.8   xfun_0.45       digest_0.6.36   rlang_1.1.4
[13] evaluate_0.24.0
```