

Félix Laplante Université de Paris Saclay

Christophe Ambroise <sup>1</sup> Laboratoire de Mathématiques et Modélisation d'Evry, Université  
Paris-Saclay, CNRS, Univ Evry,

Date published: 2025-07-06 Last modified: 2024-07-04

### Abstract

In this paper, Spectral-Bridges, a novel clustering algorithm, is introduced. This algorithm builds upon the traditional k-means and spectral clustering frameworks by subdividing data into small Voronoï regions, which are subsequently assessed for their connectivity. Drawing inspiration from Support vector machine margin concept, a non-parametric clustering approach is proposed, building an affinity margin between each pair of Voronoï regions. This approach is characterized by minimal hyperparameters and delineation of intricate, non-convex cluster structures.

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

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

## Contents

1	<b>1 Introduction</b>	<b>2</b>
3	<b>2 Related Work</b>	<b>3</b>
4	<b>3 Spectral bridges</b>	<b>3</b>
5	3.1 Bridge affinity . . . . .	4
6	3.2 Algorithm . . . . .	5
7	<b>4 Numerical experiments</b>	<b>5</b>
8	4.1 Real-world Data . . . . .	6
9	4.2 Synthetic Data . . . . .	6
10	4.2.1 Datasets Summary & Class Balance . . . . .	7
11	4.3 Metrics . . . . .	7
12	4.4 Platform . . . . .	7
13	4.5 Hyperparameter settings . . . . .	7
14	4.6 Accuracy . . . . .	7
15	4.7 Noise robustness . . . . .	10
16	<b>5 Conclusive remarks</b>	<b>11</b>

---

<sup>1</sup>Corresponding author: [christophe.ambroise@univ-evry.fr](mailto:christophe.ambroise@univ-evry.fr)

17	<b>6 Appendix</b>	<b>11</b>
18	6.1 Derivation of the bridge affinity . . . . .	11
19	<b>References</b>	<b>12</b>
20	<b>Session information</b>	<b>13</b>

## 21 1 Introduction

22 Clustering is a fundamental technique for exploratory data analysis, organizing a set of objects into  
23 distinct homogeneous groups known as clusters. It is extensively utilized across various fields, such  
24 as biology for gene expression analysis (Eisen et al. 1998), social sciences for community detection in  
25 social networks (Latouche, Birmelé, and Ambroise 2011), and psychology for identifying behavioral  
26 patterns. Clustering is often employed alongside supervised learning as a pre-processing step, helping  
27 to structure and simplify data, thus enhancing the performance and interpretability of subsequent  
28 predictive models (Verhaak et al. 2010). Additionally, clustering can be integrated into supervised  
29 learning algorithms, such as mixture of experts (Jacobs et al. 1991), as part of a multi-objective  
30 strategy.

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

33 Density-based methods identify regions within the data with a high concentration of points, corre-  
34 sponding to the modes of the joint density. A notable non-parametric example of this approach is  
35 DBSCAN (Ester et al. 1996). In contrast, model-based clustering, such as Gaussian mixture models,  
36 represents a parametric approach to density-based methods. Model-based clustering assumes that  
37 the data is generated from a mixture of underlying probability distributions, typically Gaussian  
38 distributions. Each cluster is viewed as a component of this mixture model, and the Expectation-  
39 Maximization (EM) algorithm is often used to estimate the parameters. This approach provides a  
40 probabilistic framework for clustering, allowing for the incorporation of prior knowledge and the  
41 ability to handle more complex cluster shapes and distributions (McLachlan and Peel 2000).

42 Geometric approaches, such as k-means (MacQueen et al. 1967), are distance-based methods that aim  
43 to partition data by optimizing a criterion reflecting group homogeneity. The k-means++ algorithm  
44 (Arthur and Vassilvitskii 2006) enhances this approach by providing fast, convenient, and interpretable  
45 results. However, a key limitation of these methods is the assumption of linear boundaries between  
46 clusters, implying that clusters are convex. To address non-convex clusters, the kernel trick can be  
47 applied, allowing for a more flexible k-means algorithm. This approach is comparable to spectral  
48 clustering in handling complex cluster boundaries (Dhillon, Guan, and Kulis 2004). The k-means  
49 algorithm can also be interpreted within the framework of model-based clustering under specific  
50 assumptions (Govaert and Nadif 2003), revealing that it is essentially a special case of the more  
51 general Gaussian mixture models, where clusters are assumed to be spherical Gaussian distributions  
52 with equal variance.

53 Graph-based methods represent data as a graph, with vertices symbolizing data points and edges  
54 weighted to indicate the affinity between these points. Spectral clustering can be seen as a relaxed  
55 version of the graph cut algorithm (Shi and Malik 2000). However, traditional spectral clustering faces  
56 significant limitations due to its high time and space complexity, greatly hindering its applicability  
57 to large-scale problems (Von Luxburg 2007).

58 The method we propose aims to find non-convex clusters in large datasets, without relying on  
59 parametric model, by using spectral clustering based on an affinity that characterizes the local density  
60 of the data. The algorithm described in this paper draws from numerous clustering approaches. The

initial intuition is to detect high-density areas. To this end, 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 only parameters of the algorithm are the number of Voronoi Cells and the number of clusters.

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 Related Work

Spectral clustering is a graph-based approach that computes the eigen-vectors 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). Spectral clustering enables capturing complex data structures and discerning clusters based on the connectivity of data points in a transformed space, effectively treating it as a relaxed graph cut problem.

Classical spectral clustering involves two phases: construction of the affinity matrix and eigen-decomposition. Constructing the affinity matrix requires  $O(n^2d)$  time and  $O(n)$  memory, while eigen-decomposition demands  $O(n^3)$  time and  $O(n^2)$  memory, where  $n$  is the data size and  $d$  is the dimension. As  $n$  increases, the computational load escalates significantly (Von Luxburg 2007).

To mitigate this computational burden, one common approach is to sparsify the affinity matrix and use sparse eigen-solvers, reducing memory costs but still requiring computation of all original matrix entries (Von Luxburg 2007). Another strategy is sub-matrix construction. The Nyström method randomly selects  $m$  representatives from the dataset to form an  $n \times m$  affinity sub-matrix (Chen et al. 2010). Cai et al. extended this with the landmark-based spectral clustering method, which uses k-means to determine  $m$  cluster centers as representatives (Cai and Chen 2014). Ultra-scalable spectral clustering (U-SPEC) employs a hybrid representative selection strategy and a fast approximation method for constructing a sparse affinity sub-matrix (Huang et al. 2019).

Other approaches use the properties of the small initial cluster for the affinity computation. Clustering Based on Graph of Intensity Topology (GIT) estimates a global topological graph (topo-graph) between local clusters (Gao et al. 2021). It then uses the Wasserstein Distance between predicted and prior class proportions to automatically cut noisy edges in the topo-graph and merge connected local clusters into final clusters.

The issue of characterizing the affinity between two clusters to create an edge weight is central to the efficiency of a spectral clustering algorithm operating from a submatrix.

Notice that the clustering robustness of many Spectral clustering algorithm heavily relies on the proper selection of kernel parameter, which is difficult to find without prior knowledge (Ng, Jordan, and Weiss 2001).

## 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 an “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 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 (See Figure Figure 1). If the difference is small, it suggests a low density between the classes. Conversely, if this difference 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.

Let consider sample  $X = (\mathbf{x}_i)_{i \in \{1, \dots, n\}}$  of vectors  $\mathbf{x}_i \in \mathbb{R}^p$  and a set of  $m$  coding vectors  $(\boldsymbol{\mu}_k)_{k \in \{1, \dots, m\}}$  defining a partition  $P = \{\mathcal{V}_1, \dots, \mathcal{V}_m\}$  of  $\mathbb{R}^p$  into  $m$  Voronoi regions:

$$\mathcal{V}_k = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \boldsymbol{\mu}_k\| \leq \|\mathbf{x} - \boldsymbol{\mu}_j\| \text{ for all } j \neq k\}.$$

In the following a ball denotes the subset of  $X$  in a Voronoi region. The inertia of two balls  $\mathcal{V}_k$  and  $\mathcal{V}_l$  is

$$I_{kl} = \sum_{\mathbf{x}_i \in \mathcal{V}_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 + \sum_{\mathbf{x}_i \in \mathcal{V}_l} \|\mathbf{x}_i - \boldsymbol{\mu}_l\|^2.$$

We define a bridge as a structure defined by a segment connecting two centroids  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\mu}_l$ . The inertia of a bridge between  $\mathcal{V}_k$  and  $\mathcal{V}_l$  is defined as

$$B_{kl} = \sum_{\mathbf{x}_i \in \mathcal{V}_k \cup \mathcal{V}_l} \|\mathbf{x}_i - \mathbf{p}_{kl}(\mathbf{x}_i)\|^2,$$

where

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

with

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

Considering two centroids, the normalized average of the difference between Bridge and balls inertia (See [Appendix](#)) constitutes the basis of our affinity measure between to regions:

$$\begin{aligned} \frac{B_{kl} - I_{kl}}{(n_k + n_l)\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2} &= \frac{\sum_{\mathbf{x}_i \in \mathcal{V}_k} \langle \mathbf{x}_i - \boldsymbol{\mu}_k, \boldsymbol{\mu}_l - \boldsymbol{\mu}_k \rangle_+^2 + \sum_{\mathbf{x}_i \in \mathcal{V}_l} \langle \mathbf{x}_i - \boldsymbol{\mu}_l, \boldsymbol{\mu}_k - \boldsymbol{\mu}_l \rangle_+^2}{(n_k + n_l)\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^4}, \\ &= \frac{\sum_{\mathbf{x}_i \in \mathcal{V}_k \cup \mathcal{V}_l} \alpha_i^2}{n_k + n_l}, \end{aligned}$$

where

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

The basic intuition behind this affinity is that  $t_i$  represents the relative position of the projection of  $\mathbf{x}_i$  on the segment  $[\mu_k, \mu_l]$ .  $\alpha_i$  represents the relative position on the segment, with the centroid of the class to which  $\mathbf{x}_i$  belongs as the starting point.

The boundary that separates the two clusters defined by centroids  $\mu_k$  and  $\mu_l$  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  $\mathbf{x}_i \in \mathcal{V}_k \cup \mathcal{V}_l$  which are not projected on centroids but somewhere on the segment, the distance from a point to the hyperplane is

$$\|\mathbf{p}_{kl}(\mathbf{x}_i) - \mu_{kl}\| = (1/2 - \alpha_i)\|\mu_k - \mu_l\|.$$

This distance is similar to the concept of margin in Support Vector Machine (Cortes and Vapnik 1995). When the  $\alpha_i$  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  $\alpha_i$  or  $\alpha_i^2$  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.

Eventually, we define the bridge affinity between centroids  $k$  and  $l$  as:

$$a_{kl} = \begin{cases} 0, & \text{if } k = l, \\ \frac{\sum_{\mathbf{x}_i \in \mathcal{V}_k \cup \mathcal{V}_l} \alpha_i^2}{n_k + n_l}, & \text{otherwise.} \end{cases}$$

To allow points with large margin to dominate and make the algorithm more robust to noise and outliers we consider the following exponential transformation:

$$\tilde{a}_{kl} = g(a_{kl}) = \exp \sqrt{(a_{kl})}.$$

### 3.2 Algorithm

The Spectral Bridges algorithm first identifies local clusters to define Voronoï regions, computes edges with affinity weights between these regions, and ultimately cuts edges between regions with low inter-region density to determine the final clusters (See Algorithm Algorithm 1 and Figure Figure 2).

In spectral clustering, the time complexity is usually dominated by the eigendecomposition step, which is  $O(n^3)$ . However, in the case of Spectral Bridges, the  $(k)$ -means algorithm has a time complexity of  $O(n \times m \times p)$ . For datasets with large  $(n)$ , this can be more significant than the  $O(m^3)$  time complexity of the Spectral Bridges eigendecomposition.

## 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 1** Spectral Bridges

---

```
1: procedure SPECTRALBRIDGES( $X, k, m$ )  $\triangleright$   $X$ : input dataset,  $k$ : number of clusters,  $m$ : number of  
   Voronoi regions  
2:   Step 1: Vector Quantization  
3:    $centroids, voronoiRegions \leftarrow \text{KMEANS}(X, m)$   $\triangleright$  Initial centroids and Voronoi Regions using  
   k-means++  
4:   Step 2: Affinity Computation  
5:    $A = \{g(a_{kl})\}_{kl} \leftarrow \text{AFFINITY}(X, centroids, voronoiRegions)$   
6:   Step 3: Spectral Clustering  $\triangleright$  Affect each region to a cluster  
7:    $labels \leftarrow \text{SPECTRALCLUSTERING}(A, k)$   
8:   Step 4: Propagate  $\triangleright$  Affect each data point to the cluster of its region  
9:    $clusters \leftarrow \text{PROPAGATE}(X, labels, voronoiRegions)$   
10:  return  $clusters$   $\triangleright$  Cluster labels for data points in  $X$   
11: end procedure
```

---

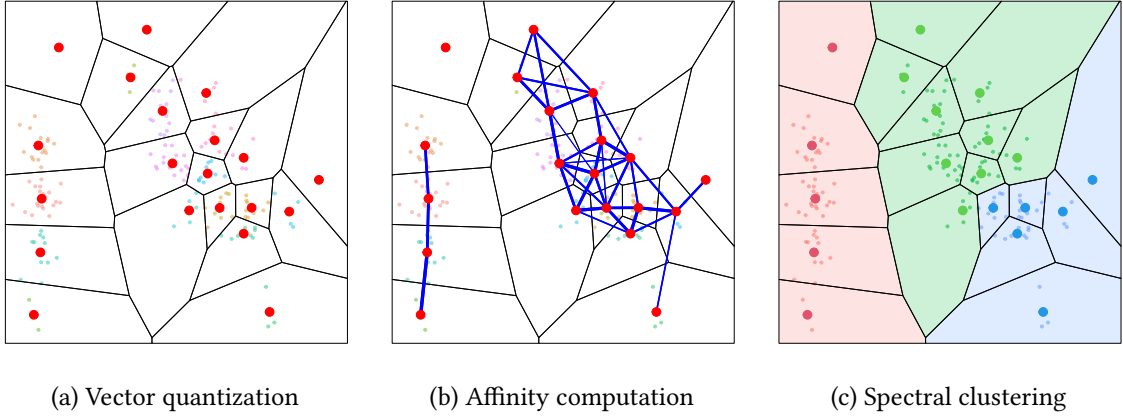


Figure 2: Illustration of the Spectral bridges algorithm with the Iris dataset (first principal plane). Vector quantization (Step 1 of Algorithm Algorithm 1 ), Affinity computation (Step 2 of Algorithm Algorithm 1 ), Spectral clustering and spreading (Step 3-4 of Algorithm Algorithm 1 ).

algorithm. We compare **Spectral-Bridges (SB)** against several state-of-the-art methods, including **k-means++ (KM)** (MacQueen et al. 1967; Arthur and Vassilvitskii 2006), **Expectation-Maximization (EM)** (Dempster, Laird, and Rubin 1977), **Ward Clustering (WC)** (Ward Jr 1963), 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.

#### 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) (Halkidi, Batistakis, and Vazirgiannis 2002) 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 Hyperparameter settings

We set the hyperparameters of our algorithm to “reasonable” values based on the size of each dataset and the number of clusters. A larger number of clusters typically suggests that a higher value for the Voronoï regions is optimal. Conversely, using a high number of Voronoï regions for a small dataset might result in nearly empty regions that do not adequately represent any local structure. We will discuss our method’s sensitivity to this parameter, but at no point did we use labels to determine its empirically optimal value.

For other algorithms, such as DBSCAN, we used labels to determine the best hyperparameter values to compare our method against the “best case scenario”, thus putting our algorithm at a voluntary disadvantage.

## 4.6 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 10 consecutive runs with the same random seed for each method. Since our computational capabilities were too limited, we sampled 20000 (one third) data points at random. This sample is not fixed and changes for each iteration.

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).

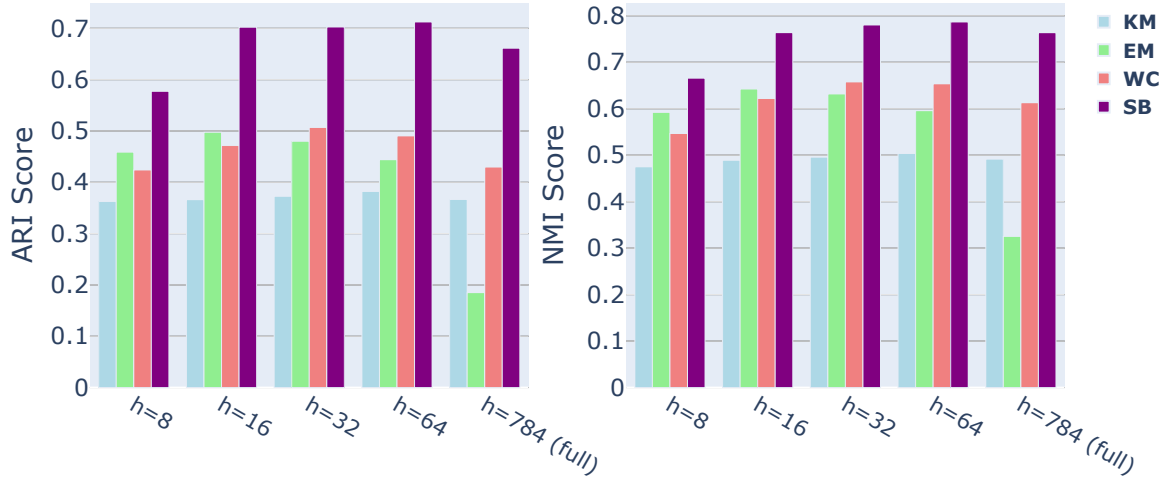


Figure 3: ARI and NMI scores of **k-means++** (blue), **EM** (green), **Ward Clustering** (red), and **Spectral-Bridges** (purple) on PCA embedding and full MNIST

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.

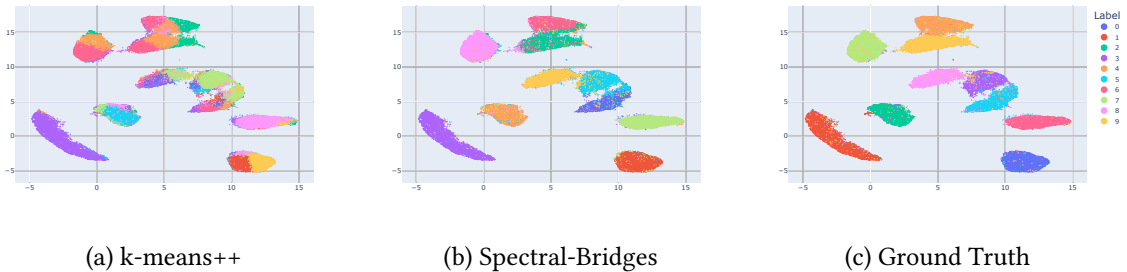
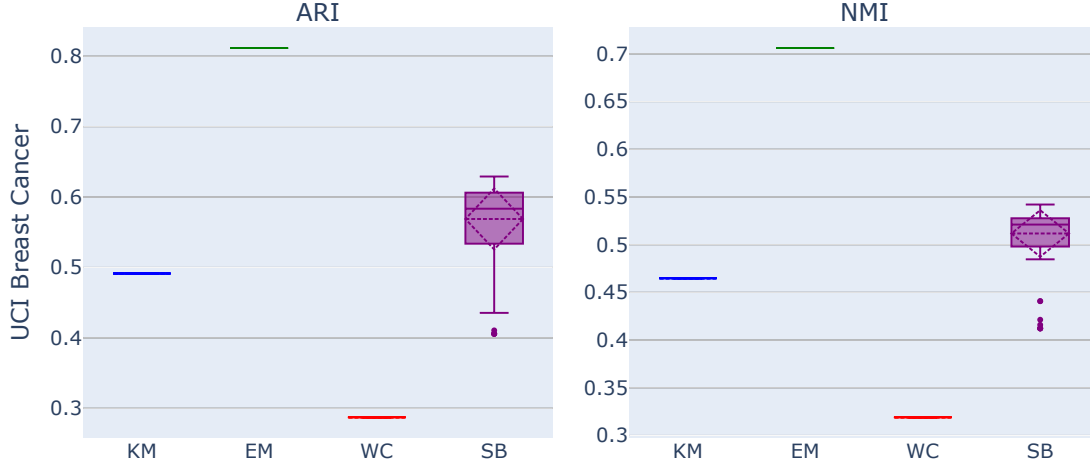


Figure 4: UMAP projection of predicted clusters against the ground truth labels.

We also put our algorithm to the test against the same competitors using scikit-learn’s



UCI Breast Cancer data. Once again, our method performs well although the advantage is not as obvious in this case. However, in none of our tests has it ranked worse than k-means++. The results are displayed as a boxplot generated from 200 iterations of each algorithm using a different seed, in order to better grasp the variability lying in the seed dependent nature of the k-means++, Expectation Maximization and Spectral-Bridges algorithms.



Since we expect our algorithm to shine at discerning complex and intricate cluster structures, we collected an array four toy datasets illustrated below.

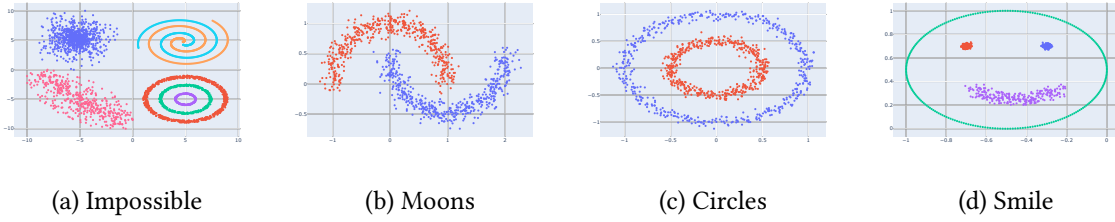


Figure 5: Visualization of the four toy datasets.

We once again benchmark multiple algorithms including ours in the exact same manner as for the UCI Breast Cancer data. Results show our method outperforms all tested algorithms (DBSCAN, k-means++, Expectation Maximization and Ward Clustering) while needing few hyperparameters. In this regard, as discussed previously, DBSCAN's parameters have been set using the labels to illustrate a best case scenario, although, in practice, one can expect worse results. Despite this, Spectral-Bridge still manages to better encompass the clusters' structure.

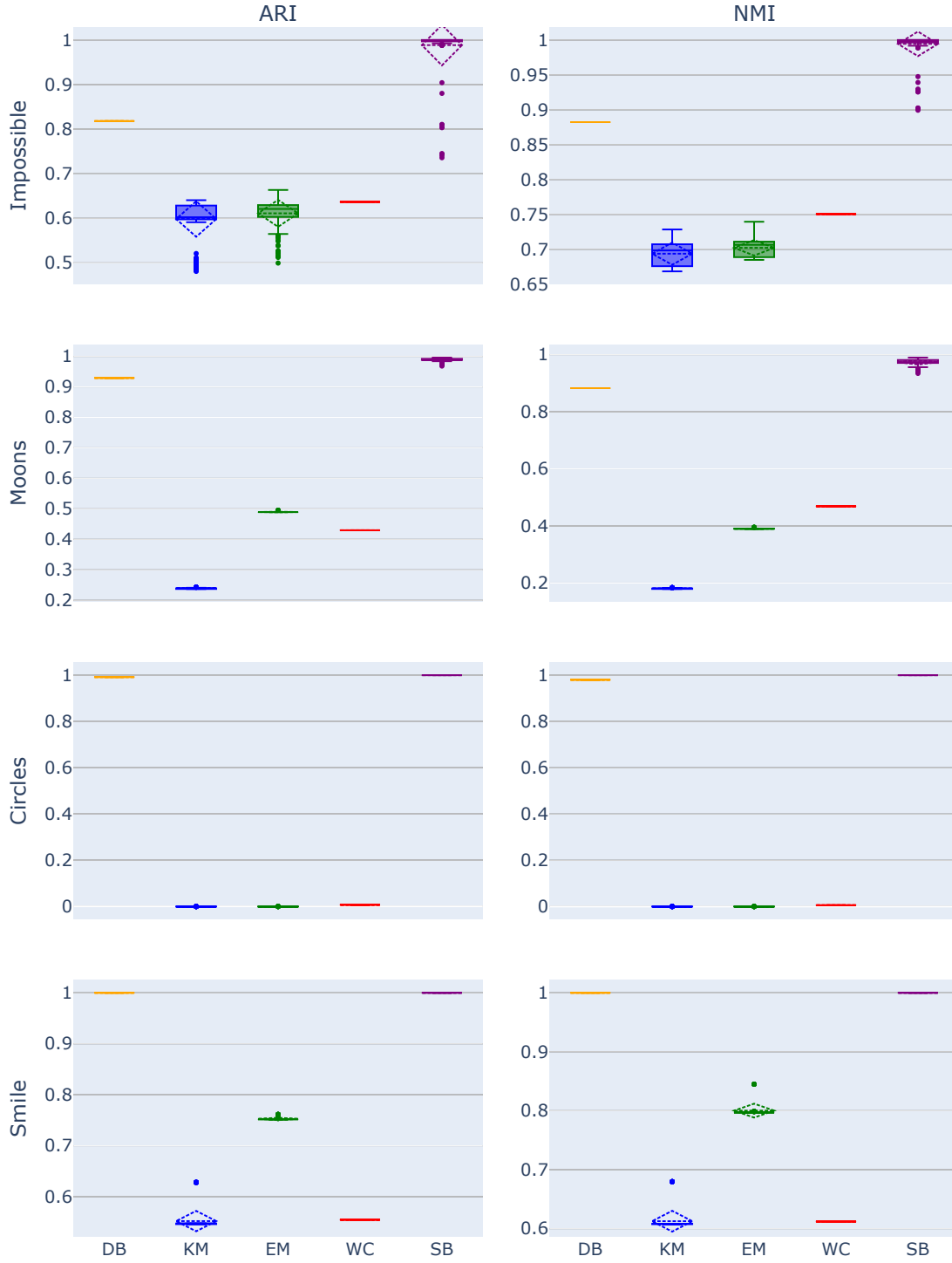


Figure 6: ARI and NMI scores of Spectral-Bridges and competitors on standard synthetic toy datasets

#### 4.7 Noise robustness

To evaluate the noise robustness of our algorithm, we imagined two setups : introducing Gaussian distributed perturbations and concatenating uniformly distributed points on a predefined rectangular

region (whose size is determined by the span of the dataset) to a given dataset. As illustrated below, in both of these cases, our tests show our method is quite insensitive to noise in those scenarios.

Here are three representations of our algorithm's predicted cluster centers displayed as colored dots, while the each point of the Impossible dataset is shown as a small black dot. In the first graph, the dataset is not modified. In the second, we added  $n = 250$  uniformly distributed samples and in the last we used Gaussian noise perturbations with  $\sigma = 0.1$ .

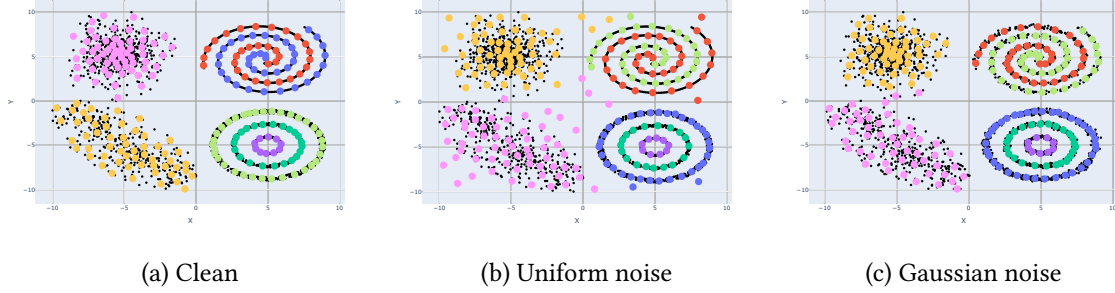


Figure 7: Effects of noise on Spectral-Bridges's output.

## 5 Conclusive remarks

Possibility to kernelize

## 6 Appendix

### 6.1 Derivation of the bridge affinity

We denote a bridge as a segment connecting two centroids  $\mu_k$  and  $\mu_l$ . The inertia of a bridge between  $\mathcal{V}_k$  and  $\mathcal{V}_l$  is defined as

$$B_{kl} = \sum_{\mathbf{x}_i \in \mathcal{V}_k \cup \mathcal{V}_l} \|\mathbf{x}_i - \mathbf{p}_{kl}(\mathbf{x}_i)\|^2,$$

where

$$\mathbf{p}_{kl}(\mathbf{x}_i) = \mu_k + t_i(\mu_l - \mu_k),$$

with

$$t_i = \min \left( 1, \max \left( 0, \frac{\langle \mathbf{x}_i - \mu_k | \mu_l - \mu_k \rangle}{\|\mu_l - \mu_k\|^2} \right) \right).$$

$B_{kl}$ , the bridge inertia between centroids  $k$  and  $l$ , can be expressed as the sum of three terms, which represents the projection onto each centroids and onto the segment:

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

The last term may be decomposed in two parts corresponding to the points of the two Voronoi regions which are projected on the segment:

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

247 and each part further decomposed using Pythagore

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

248 Thus

$$\begin{aligned}
B_{kl} - I_{kl} &= \sum_{i|t_i \in ]0, \frac{1}{2}[} t_i^2 \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2 + \sum_{i|t_i \in ]\frac{1}{2}, 1[} (1 - t_i)^2 \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2, \\
\frac{B_{kl} - I_{kl}}{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2} &= \sum_{i|t_i \in ]0, \frac{1}{2}[} t_i^2 + \sum_{i|t_i \in ]\frac{1}{2}, 1[} (1 - t_i)^2, \\
\frac{B_{kl} - I_{kl}}{(n_k + n_l)\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2} &= \frac{\sum_{i \in k} \langle \mathbf{x}_i - \boldsymbol{\mu}_k | \boldsymbol{\mu}_l - \boldsymbol{\mu}_k \rangle_+^2 + \sum_{i \in l} \langle \mathbf{x}_i - \boldsymbol{\mu}_l | \boldsymbol{\mu}_k - \boldsymbol{\mu}_l \rangle_+^2}{(n_k + n_l)\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^4}.
\end{aligned}$$

## 249 References

- 250 Arthur, David, and Sergei Vassilvitskii. 2006. “K-Means++: The Advantages of Careful Seeding.”  
251 Technical Report 2006-13. Stanford InfoLab; Stanford. <http://ilpubs.stanford.edu:8090/778/>.
- 252 Cai, Deng, and Xinlei Chen. 2014. “Large Scale Spectral Clustering via Landmark-Based Sparse  
253 Representation.” *IEEE Transactions on Cybernetics* 45 (8): 1669–80.
- 254 Chen, Wen-Yen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y Chang. 2010. “Parallel  
255 Spectral Clustering in Distributed Systems.” *IEEE Transactions on Pattern Analysis and Machine  
256 Intelligence* 33 (3): 568–86.
- 257 Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” *Machine Learning* 20 (3):  
258 273–97.
- 259 Dempster, Arthur P, Nan M Laird, and Donald B Rubin. 1977. “Maximum Likelihood from Incomplete  
260 Data via the EM Algorithm.” *Journal of the Royal Statistical Society: Series B (Methodological)* 39  
261 (1): 1–22.
- 262 Dhillon, Inderjit S, Yuqiang Guan, and Brian Kulis. 2004. “Kernel k-Means, Spectral Clustering  
263 and Normalized Cuts.” In *Proceedings of the Tenth ACM SIGKDD International Conference on  
264 Knowledge Discovery and Data Mining*, 551–56. ACM.
- 265 Eisen, Michael B., Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. “Cluster Analysis  
266 and Display of Genome-Wide Expression Patterns.” *Proceedings of the National Academy of  
267 Sciences* 95 (25): 14863–68.
- 268 Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. “A Density-Based Algorithm  
269 for Discovering Clusters in Large Spatial Databases with Noise.” In *Kdd*, 96:226–31.
- 270 Gao, Zhangyang, Haitao Lin, Cheng Tan, Lirong Wu, Stan Li, et al. 2021. “Git: Clustering Based on  
271 Graph of Intensity Topology.” *arXiv Preprint arXiv:2110.01274*.
- 272 Govaert, Gérard, and Mohamed Nadif. 2003. “Clustering with Block Mixture Models.” *Pattern  
273 Recognition* 36 (2): 463–73.
- 274 Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis. 2002. “Cluster Validity Methods: Part i.”  
275 *ACM SIGMOD Record* 31 (2): 40–45.

276 Huang, Dong, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh. 2019.  
 277 “Ultra-Scalable Spectral Clustering and Ensemble Clustering.” *IEEE Transactions on Knowledge*  
 278 *and Data Engineering* 32 (6): 1212–26.

279 Jacobs, Robert A, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. “Adaptive Mixtures  
 280 of Local Experts.” *Neural Computation* 3 (1): 79–87.

281 Latouche, Pierre, Etienne Birmelé, and Christophe Ambroise. 2011. “Overlapping stochastic block  
 282 models with application to the French political blogosphere.” *The Annals of Applied Statistics* 5  
 283 (1): 309–36. <https://doi.org/10.1214/10-AOAS382>.

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

287 McLachlan, Geoffrey J., and David Peel. 2000. *Finite Mixture Models*. New York: Wiley-Interscience.

288 Ng, Andrew, Michael Jordan, and Yair Weiss. 2001. “On Spectral Clustering: Analysis and an  
 289 Algorithm.” *Advances in Neural Information Processing Systems* 14.

290 Shi, Jianbo, and Jitendra Malik. 2000. “Normalized Cuts and Image Segmentation.” *IEEE Transactions*  
 291 *on Pattern Analysis and Machine Intelligence* 22 (8): 888–905.

292 Verhaak, Roel G. W., Katherine A. Hoadley, Elizabeth Purdom, Victoria Wang, Yuexin Qi, Matthew  
 293 D. Wilkerson, Charlie R. Miller, et al. 2010. “Integrated Genomic Analysis Identifies Clinically  
 294 Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR,  
 295 and NF1.” *Cancer Cell* 17 (1): 98–110.

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

297 Ward Jr, Joe H. 1963. “Hierarchical Grouping to Optimize an Objective Function.” *Journal of the*  
 298 *American Statistical Association* 58 (301): 236–44.

## 299 Session information

```
300 R version 4.3.2 (2023-10-31)
301 Platform: x86_64-apple-darwin20 (64-bit)
302 Running under: macOS Sonoma 14.3.1
303
304 Matrix products: default
305 BLAS: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
306 LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib; LAPACK
307
308 locale:
309 [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
310
311 time zone: Europe/Paris
312 tzcode source: internal
313
314 attached base packages:
315 [1] stats graphics grDevices utils datasets methods base
316
317 loaded via a namespace (and not attached):
318 [1] compiler_4.3.2 fastmap_1.1.1 cli_3.6.2 tools_4.3.2
319 [5] htmltools_0.5.7 rstudioapi_0.15.0 yaml_2.3.8 rmarkdown_2.26
320 [9] knitr_1.45 jsonlite_1.8.8 xfun_0.42 digest_0.6.34
321 [13] rlang_1.1.3 evaluate_0.23
```