COMPUTO

ISSN 2824-7795

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,

Date published: 2025-07-06 Last modified: 2024-06-19

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 Ward linkage, a non-parametric clustering approach is embraced by the Spectral-Bridges algorithm. 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. K-means centroids are leveraged for subgroup initialization, and a strategy is proposed to link these regions, with the "reward" gauged in terms of newly captured variance achievable by connecting them through a projected data segment, referred to as a bridge.

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.

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

3.1 Bridge gain

The affinity matrix Let $A = (a_{kl})_{1\leq k,l\leq n}$ be the affinity matrix:

$$a_{kl} = \Big[\frac{\sum_{\boldsymbol{x}\in P_k} \mathrm{d}^2(\boldsymbol{x},\boldsymbol{\mu}_k) + \sum_{\boldsymbol{x}\in P_l} \mathrm{d}^2(\boldsymbol{x},\boldsymbol{\mu}_l) - \sum_{\boldsymbol{x}\in P_k\cup P_l} \mathrm{d}^2(\boldsymbol{x},[\boldsymbol{\mu}_k,\boldsymbol{\mu}_l])}{(\#P_k+\#P_l)\|\boldsymbol{\mu}_k-\boldsymbol{\mu}_l\|^2}\Big]^{1/2}$$

One can rewrite this :

$$a_{kl} = \Big[\frac{\sum_{\boldsymbol{x}\in P_k}\langle \boldsymbol{x}-\boldsymbol{\mu}_k|\boldsymbol{\mu}_l-\boldsymbol{\mu}_k\rangle_+ + \sum_{\boldsymbol{x}\in P_l}\langle \boldsymbol{x}-\boldsymbol{\mu}_l|\boldsymbol{\mu}_k-\boldsymbol{\mu}_l\rangle_+}{(\#P_k+\#P_l)\|\boldsymbol{\mu}_k-\boldsymbol{\mu}_l\|^2}\Big]^{1/2}$$

Because for each $\boldsymbol{x}\in P_k$, let us denote $\boldsymbol{x}_\perp$ the orthogonal projection on the right line : $(\boldsymbol{\mu}_k,\boldsymbol{\mu}_l)$.

- If $\boldsymbol{x}_\perp \notin [\boldsymbol{\mu}_k,\boldsymbol{\mu}_l]$, *i.e.* $\langle \boldsymbol{x}-\boldsymbol{\mu}_k|\boldsymbol{\mu}_l-\boldsymbol{\mu}_k\rangle < 0$, then the point $\boldsymbol{x}$ is closest to $\boldsymbol{\mu}_k$. In that case, the difference between $\mathrm{d}^2(\boldsymbol{x},\boldsymbol{\mu}_k) - \mathrm{d}^2(\boldsymbol{x},[\boldsymbol{\mu}_k,\boldsymbol{\mu}_l]) = 0$.

- If $\boldsymbol{x}_\perp \in [\boldsymbol{\mu}_k, \boldsymbol{\mu}_l]$, *i.e.* $\langle \boldsymbol{x} - \boldsymbol{\mu}_k | \boldsymbol{\mu}_l - \boldsymbol{\mu}_k \rangle \geq 0$. And, by the Pythagorean theorem, we have : $\|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2 = \|\boldsymbol{x} - \boldsymbol{x}_\perp\|^2 + \|\boldsymbol{x}_\perp - \boldsymbol{\mu}_k\|^2$, so $\setminus \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2 - \|\boldsymbol{x} - \boldsymbol{x}_\perp\|^2 = \|\boldsymbol{x}_\perp - \boldsymbol{\mu}_k\|^2$ and $\mathrm{d}^2(\boldsymbol{x}, \boldsymbol{\mu}_k) - \mathrm{d}^2(\boldsymbol{x}, [\boldsymbol{\mu}_k, \boldsymbol{\mu}_l]) = \|\boldsymbol{x}_\perp - \boldsymbol{\mu}_k\|^2 = \langle \boldsymbol{x} - \boldsymbol{\mu}_k, \boldsymbol{\mu}_l - \boldsymbol{\mu}_k \rangle$

Thus, more concisely, $\forall \boldsymbol{x} \in P_k$, one can write :

$$\mathrm{d}^2(\boldsymbol{x}, \boldsymbol{\mu}_k) - \mathrm{d}^2(\boldsymbol{x}, [\boldsymbol{\mu}_k, \boldsymbol{\mu}_l]) = \langle \boldsymbol{x} - \boldsymbol{\mu}_k | \boldsymbol{\mu}_l - \boldsymbol{\mu}_k \rangle_+$$

3.2 Algorithm

4 Numerical experiments

References

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 4.3.2 (2023-10-31)
Platform: x86_64-apple-darwin20 (64-bit)
Running under: macOS Sonoma 14.3.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPAC

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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_4.3.2    fastmap_1.1.1     cli_3.6.2         tools_4.3.2
 [5] htmltools_0.5.7   rstudioapi_0.15.0 yaml_2.3.8        rmarkdown_2.26
 [9] knitr_1.45        jsonlite_1.8.8    xfun_0.42         digest_0.6.34
[13] rlang_1.1.3       evaluate_0.23
```

Algorithm 1 Spectral Bridges

procedure SpectralBridges(*data*, *k*, *p*) ▷ data: input dataset, k: number of clusters, p: number of Voronoi regions
    centroids ← KMeans(*data*, *p*) ▷ Initial centroids using k-means
    voronoiRegions ← Subdivide(*data*, *centroids*) ▷ Subdivide data into Voronoi regions
    graph ← CreateGraph(*voronoiRegions*) ▷ Assess connectivity between regions
    clusters ← WardLinkage(*graph*, *k*) ▷ Cluster using Ward linkage-inspired approach
    return *clusters*
end procedure
procedure KMeans(*data*, *p*)
    Initialize *p* centroids randomly
    repeat
        Assign each point to the nearest centroid
        Update centroids based on assignments
    until centroids do not change
    return centroids
end procedure
procedure Subdivide(*data*, *centroids*)
    voronoiRegions ← {}
    for each point *x* in *data* **do**
        Find the nearest centroid for *x*
        Assign *x* to the corresponding Voronoi region
    end for
    return *voronoiRegions*
end procedure
procedure CreateGraph(*voronoiRegions*)
    graph ← empty graph
    for each pair of regions (R_i, R_j) in *voronoiRegions* **do**
        Calculate connectivity measure between R_i and R_j
        Add edge between R_i and R_j in *graph* with weight based on connectivity
    end for
    return *graph*
end procedure
procedure WardLinkage(*graph*, *k*)
    clusters ← Initialize each region as a separate cluster
    repeat
        Find the pair of clusters with the smallest merging cost
        Merge the selected pair of clusters
    until number of clusters equals *k*
    return *clusters*
end procedure