spectral clustering

Author and editors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

Spectral clustering is a graph clustering method that computes the clusters of a dataset from the eigenvectors of the Laplacian matrix of a graph whose nodes are the data points and whose edge weights express their similarity. The Laplacian matrix is positive semi-definite (psd), and the multiplicity of its eigenvalue 0 equals the number of connected components of the graph, with their indicator vectors as eigenvectors. For k clusters, the entries of the k eigenvectors with the smallest eigenvalues form a new feature vector for each node, and k-means on these feature vectors delivers the clusters. For two clusters, the method thresholds the Fiedler vector. Spectral clustering is a relaxation of the ratio cut, which divides the total weight of the edges between the clusters by their sizes, and the second-smallest eigenvalue is a lower bound on the ratio cut of every bipartition. Unlike k-means on the raw feature vectors, spectral clustering finds clusters that are connected in the graph but not convex in the feature space. The number of clusters and the construction of the graph are its hyperparameters.

Definition

Each day of 2024 at the weather station Krems an der Donau is a data point with a feature vector $\mathbf{x} \in \mathbb{R}^2$ that holds the morning minimum and the maximum daytime temperature of the day. Connecting each day with its ten nearest days, measured by the Euclidean distance between their feature vectors, turns the 366 days into the nodes $i = 1, \ldots, n$ of a k-nearest-neighbor graph (k-NN graph) $\mathcal{G}$ with $n = 366$ nodes and 2209 edges (Fig. 2(a)). A cluster of this graph is a set of days that are joined by many edges among themselves and by few edges to the remaining days. Spectral clustering is a graph clustering method that finds such clusters from the eigenvectors of the Laplacian matrix of the graph (Hastie et al., 2009, Sect. 14.5.3; von Luxburg, 2007; Ng et al., 2001; Shi and Malik, 2000).

Consider a similarity graph of the dataset: a weighted graph $\mathcal{G}$ with nodes $i = 1, \ldots, n$, one node per data point, and edge weights $A_{i,i'} = A_{i',i} \geq 0$ that express the similarity between the data points i and i'; a missing edge has weight 0. The edge weights form the adjacency matrix $\mathbf{A}^{(\mathcal{G})} \in \mathbb{R}^{n \times n}$, and the node degree $d^{(i)} = \sum_{i'=1}^{n} A_{i,i'}$ of node i is the sum of its edge weights. The

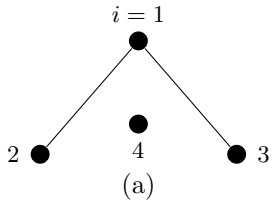
Laplacian matrix of $\mathcal{G}$ is

$$\mathbf{L}^{(\mathcal{G})} := \mathbf{D} - \mathbf{A}^{(\mathcal{G})} \quad \text{with } \mathbf{D} := \operatorname{diag}\left\{d^{(1)}, \ldots, d^{(n)}\right\}. \tag{1}$$

For every vector $\mathbf{u} \in \mathbb{R}^n$ with entries u_i, the Laplacian matrix (1) satisfies

$$\mathbf{u}^\top \mathbf{L}^{(\mathcal{G})} \mathbf{u} = \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} A_{i,i'} \left(u_i - u_{i'}\right)^2, \tag{2}$$

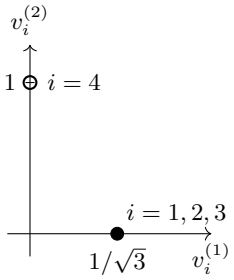
a weighted sum of squared differences across the edges (von Luxburg, 2007, Prop. 1). The quadratic form (2) is never negative, so $\mathbf{L}^{(\mathcal{G})}$ is a symmetric matrix that is positive semi-definite (psd). It has an eigenvalue decomposition (EVD) $\mathbf{L}^{(\mathcal{G})} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ with an orthogonal matrix $\mathbf{V} = \left(\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\right)$ of eigenvectors and with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. The constant vector is an eigenvector for the eigenvalue 0, since every row of $\mathbf{L}^{(\mathcal{G})}$ sums to zero. More generally, the multiplicity of the eigenvalue 0 is the number of connected components of $\mathcal{G}$, and the indicator vectors of the components span its eigenvectors (von Luxburg, 2007, Prop. 2). The reason is the quadratic form (2): $\mathbf{L}^{(\mathcal{G})}\mathbf{u} = \mathbf{0}$ forces $u_i = u_{i'}$ across every edge, so $\mathbf{u}$ is constant on each component. Fig. 1 shows an undirected graph with two components; its two eigenvectors for the eigenvalue 0 place the nodes of each component on a single point of a scatterplot.



(a)

$$\mathbf{L}^{(\mathcal{G})} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

(b)



(c)

$$\mathbf{v}^{(1)} = \tfrac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{v}^{(2)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

(d)

Figure 1: (a) An undirected graph $\mathcal{G}$ with four nodes $i = 1, 2, 3, 4$ and unit edge weights, each node a data point; node 4 has no edge, so the graph has two connected components. (b) Its Laplacian matrix $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{4\times 4}$, whose eigenvalue 0 has multiplicity two. (c) Scatterplot of the nodes using the new feature vectors $\widetilde{\mathbf{x}}^{(i)} = \left(v_i^{(1)}, v_i^{(2)}\right)^\top$: the nodes 1, 2, 3 of the first component share one point (filled circle) and node 4 sits at another (open circle). (d) The two eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathbb{R}^4$ for the eigenvalue 0, the normalized indicator vectors of the two components

A graph whose clusters are joined by few edges of small weight is a perturbation of a graph with several components. The eigenvectors of its smallest eigenvalues are then a perturbation of the indicator vectors (von Luxburg, 2007, Sect. 7). Spectral clustering with k clusters therefore uses these eigenvectors as new features: node i is represented by the feature vector

$$\widetilde{\mathbf{x}}^{(i)} := \left(v_i^{(1)}, \ldots, v_i^{(k)}\right)^\top \in \mathbb{R}^k, \tag{3}$$

which holds the ith entries of the k eigenvectors with the smallest eigenvalues. k-means is then applied to the feature vectors $\widetilde{\mathbf{x}}^{(1)}, \ldots, \widetilde{\mathbf{x}}^{(n)}$ (von Luxburg, 2007, Sect. 4). Algorithm 1 states the method in full.

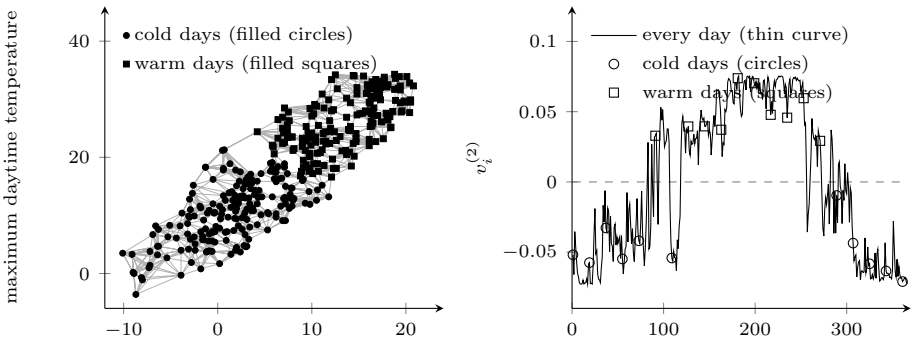
Algorithm 1 Spectral clustering (von Luxburg, 2007, Sect. 4)

Input: weighted graph $\mathcal{G}$ with nodes $i = 1, \ldots, n$ and edge weights $A_{i,i'}$, cluster count k

1: compute the Laplacian matrix (1) of $\mathcal{G}$
2: compute eigenvectors $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(k)}$ of $\mathbf{L}^{(\mathcal{G})}$ for its k smallest eigenvalues
3: form the feature vectors (3) for $i = 1, \ldots, n$
4: run k-means with k clusters on $\widetilde{\mathbf{x}}^{(1)}, \ldots, \widetilde{\mathbf{x}}^{(n)}$

Output: the cluster of each node i, i.e., of each data point

For $k = 2$ the first eigenvector is constant, so the new feature of node i is its entry $v_i^{(2)}$ in the Fiedler vector, the eigenvector of the algebraic connectivity λ_2 (Fiedler, 1973). The Fiedler vector is orthogonal to the constant vector, so it has entries of both signs, and k-means on its entries amounts to a threshold. On the days at Krems, the graph is connected with $\lambda_2 = 0.044$ and $\lambda_3 = 0.179$, and the Fiedler vector has 190 negative and 176 positive entries (Fig. 2(b)). k-means on these entries delivers 192 cold days and 174 warm days. Every day of January, February, November, and December is a cold day, and every day from May to August is a warm day. Each of the two clusters is connected in the graph, and the two are joined by 25 of the 2209 edges. k-means on the raw feature vectors $\mathbf{x}^{(i)}$, whose clusters are always convex regions of the feature space, delivers almost the same hard clustering: the two methods disagree on two days of October only. The methods differ once the clusters are not convex. Data points on two concentric rings are joined by the edges of a k-NN graph along each ring; spectral clustering returns the rings, while k-means on the raw feature vectors cuts through them (Hastie et al., 2009, Sect. 14.5.3).



(a) the 10-nearest-neighbor graph of the 366 days

(b) entries of the Fiedler vector $\mathbf{v}^{(2)}$

Figure 2: (a) The 366 days of 2024 at Krems, each a node drawn at its feature vector, and the 2209 edges (thin gray lines) of the 10-nearest-neighbor graph; all days are drawn, since the connectivity of the graph is what the panel shows. The two clusters found by spectral clustering with $k = 2$ are the cold days (filled circles) and the warm days (filled squares). (b) The entry $v_i^{(2)}$ of the Fiedler vector for every day (thin curve) against the day of the year, with every 18th day, 21 days in all, marked as a cold day (circles) or a warm day (squares); the dashed line is the zero level. Data generated by `pythondemos/specclustering.py`

Spectral clustering approximately solves a graph partitioning problem (von Luxburg, 2007, Sect. 5). For a subset $\mathcal{C} \subseteq \mathcal{V}$ of the nodes, the cut $\mathrm{cut}(\mathcal{C}) := \sum_{i\in\mathcal{C}} \sum_{i'\notin\mathcal{C}} A_{i,i'}$ is the total weight of the edges leaving $\mathcal{C}$. The subset with the smallest cut is often a single node, so the ratio cut divides the cut by the sizes of both parts (Hagen and Kahng, 1992; von Luxburg, 2007, Sect. 5),

$$\mathrm{RatioCut}(\mathcal{C}) := \mathrm{cut}(\mathcal{C})\left(\frac{1}{|\mathcal{C}|} + \frac{1}{|\mathcal{V}\setminus\mathcal{C}|}\right). \tag{4}$$

Minimizing the ratio cut (4) over all subsets $\mathcal{C}$ is NP-hard (von Luxburg, 2007, Sect. 5.1). The vector $\mathbf{u}$ with entries $u_i = \sqrt{|\mathcal{V}\setminus\mathcal{C}|/|\mathcal{C}|}$ for $i \in \mathcal{C}$ and $u_i = -\sqrt{|\mathcal{C}|/|\mathcal{V}\setminus\mathcal{C}|}$ otherwise satisfies $\mathbf{u}^\top \mathbf{L}^{(\mathcal{G})}\mathbf{u} = n\,\mathrm{RatioCut}(\mathcal{C})$ by the quadratic form (2), as well as $\mathbf{u}^\top\mathbf{1} = 0$ and $\|\mathbf{u}\|_2^2 = n$ (von Luxburg, 2007, Sect. 5.1). Dropping the requirement that $\mathbf{u}$ takes only these two values leaves the relaxed problem

$$\min_{\mathbf{u}\in\mathbb{R}^n} \mathbf{u}^\top\mathbf{L}^{(\mathcal{G})}\mathbf{u} \quad \text{subject to } \mathbf{u}^\top\mathbf{1} = 0 \text{ and } \|\mathbf{u}\|_2^2 = n, \tag{5}$$

and the relaxed problem (5) has the minimum $n\lambda_2$, attained at $\sqrt{n}\,\mathbf{v}^{(2)}$ by the Rayleigh–Ritz theorem (von Luxburg, 2007, Sect. 5.1). Thresholding the Fiedler vector turns this minimizer back into a subset, and every subset has a ratio cut of at least λ_2. On the days at Krems, the spectral bipartition has a ratio cut of 0.274, the bipartition of k-means on the raw feature vectors one of 0.383 (it cuts 35 edges), and a random bipartition one of 12.5, against $\lambda_2 = 0.044$. For

general k, the same relaxation of the ratio cut leads to the first k eigenvectors and the feature vector (3) (von Luxburg, 2007, Sect. 5.2). Replacing $|\mathcal{C}|$ by the sum of the node degrees in $\mathcal{C}$ gives the normalized cut (Shi and Malik, 2000), whose relaxation leads to the eigenvectors of the normalized Laplacian matrix $\mathbf{D}^{-1}\mathbf{L}^{(\mathcal{G})}$ (von Luxburg, 2007, Sect. 5.3). A variant uses the eigenvectors of $\mathbf{D}^{-1/2}\mathbf{L}^{(\mathcal{G})}\mathbf{D}^{-1/2}$ and scales each feature vector (3) to unit norm (Ng et al., 2001).

Spectral clustering is not a fixed-point iteration: the EVD is computed once, and iterations occur only in the k-means step. That step is empirical risk minimization (ERM) with the clustering error measured in the feature vectors (3) rather than in the raw ones. The learned hypothesis assigns a cluster to each node of the given graph only; a new data point needs a new node, a new Laplacian matrix, and a new EVD. The hyperparameters are the number k of clusters and the construction of the graph: how many neighbors the k-NN graph connects, or the bandwidth of a Gaussian kernel that turns distances into edge weights (von Luxburg, 2007, Sect. 8.1). A gap in the spectrum, as between $\lambda_2 = 0.044$ and $\lambda_3 = 0.179$ at Krems, is the usual informal guide to k (von Luxburg, 2007, Sect. 8.3); with $k = 3$, the days at Krems split into 107 cold, 170 mild, and 89 warm days, each cluster connected in the graph.

Spectral clustering is used for image segmentation, where the nodes are pixels or patches and the edge weights decrease with the difference in color and in position (Shi and Malik, 2000), and for community detection in social networks, where the edges are given and no feature vector is needed (Fortunato, 2010).

Cross References

clustering, graph clustering, cluster, k-means, clustering error, hard clustering, Laplacian matrix, adjacency matrix, node degree, EVD, eigenvalue, eigenvector, Fiedler vector, algebraic connectivity, connected, psd, k-NN graph, similarity graph, weighted graph, edge weight, image segmentation, hyperparameter

References

1. Fiedler (1973). Algebraic connectivity of graphs. Czechoslovak Mathematical Journal, 23(2), pp. 298–305.

2. Fortunato (2010). Community detection in graphs. Physics Reports, 486(3–5), pp. 75–174.

3. Hagen and Kahng (1992). New Spectral Methods for Ratio Cut Partitioning and Clustering. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 11(9), pp. 1074–1085.

4. von Luxburg (2007). A tutorial on spectral clustering. Statist. Comput., 17(4), pp. 395–416.

5. Ng et al. (2001). On Spectral Clustering: Analysis and an Algorithm. In: Adv. Neural Inf. Process. Syst., pp. 849–856.

6. Shi and Malik (2000). Normalized Cuts and Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8), pp. 888–905.

7. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.