Spectral Clustering

Ilias S. Kotsireas

Director, CARGO Lab

ikotsire@wlu.ca

www.cargo.wlu.ca

Table of Contents

Ulrike von Luxburg
A tutorial on spectral clustering.
Stat. Comput. 17 (2007), no. 4, pp. 395-416.

Introduction & Motivation

- Clustering is one of the most widely used techniques for exploratory data analysis, with applications ranging from statistics, computer science, biology to social sciences or psychology
- In virtually every scientific field dealing with empirical data, people attempt to get a first impression on their data by trying to identify groups of similar behavior in their data
- Traditional clustering algorithms such as k-means can often fail to identify (satisfactory) clusters
- Spectral clustering algorithms often outperform the traditional approaches
- Spectral clustering can be implemented efficiently by standard Linear Algebra methods

Similarity graphs

- Given a set of data points $x_1, \ldots, x_n$ and some notion of similarity $s_{ij} \geq 0$ between all pairs of data points x_i and x_j , the intuitive goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other
- If we do not have more information than similarities between data points, a nice way of representing the data is in the form of the similarity graph $G = (V, E)$
- Each vertex v_i in this graph represents a data point x_i
- Two vertices are connected if the similarity s_{ij} between the corresponding data points x_i and x_j is positive or larger than a certain threshold, and the edge is weighted by s_{ij}.
- The problem of clustering can now be reformulated using the similarity graph: we want to find a partition of the graph such that the edges between different groups have very low weights (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weights (which means that points within the same cluster are similar to each other)
- To formalize this intuition: (1) introduce some basic graph notation (2) briefly discuss the kind of graphs we use

Graph notations

- $G = (V, E)$ is an undirected graph with vertex set $V = \{v_1, \ldots, v_n\}$
- assume G is weighted: each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geq 0$
- The weighted adjacency matrix of the graph is the $n \times n$ matrix $W = (w_{ij})$
- If $w_{ij} = 0$ then the vertices v_i and v_j are not connected by an edge
- G is undirected: we require $w_{ij} = w_{ji}$
- The degree of a vertex $v_i \in V$ is: $$d_i = \sum_{j=1}^{n} w_{ij}$$
- (the sum only runs over all vertices adjacent to v_i: for all other vertices v_j the weight w_{ij} is 0)
- The degree matrix D is defined as the diagonal matrix with the degrees $d_1, \ldots, d_n$ on the diagonal

- Given a subset of vertices $A \subset V$, we denote its complement $V - A$ by $\bar{A}$
- The indicator vector $\mathbb{1}_A = (f_1, \ldots, f_n)^t \in \mathbb{R}^n$ as the vector with entries $f_i = 1$ if $v_i \in A$ and $f_i = 0$ otherwise
- shorthand notation: $i \in A$ for the set of indices $\{i | v_i \in A\}$
- For two (not necessarily disjoint) sets $A, B \subset V$ we define $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$
- Two different ways of measuring the "size" of a subset $A \subset V$:
  - $|A| = \#$ of vertices in A
  - $vol(A) = \sum_{i \in A} d_i$

    (sum over the weights of all edges attached to vertices in A)
- A subset $A \subset V$ of a graph is connected if any two vertices in A can be joined by a path such that all intermediate points also lie in A
- A subset A is called a connected component if it is connected and if there are no connections between vertices in A and $\bar{A}$
- The nonempty sets $A_1, \ldots, A_k$ form a partition of the graph, if $A_i \cap A_j =$ and $A_1 \cup \ldots \cup A_k = V$

Important similarity graphs

- There are several popular constructions to transform a given set $x_1, \ldots, x_n$ of data points with pairwise similarities s_{ij} or pairwise distances d_{ij} into a graph
- When constructing similarity graphs the goal is to model the local neighborhood relationships between the data points

(1) ϵ-neighborhood graph

Connect all points whose pairwise distances are smaller than ϵ

As the distances between all connected points are roughly of the same scale (at most ϵ), weighting the edges would not incorporate more information about the data to the graph

Therefore, the ϵ-neighborhood graph is usually considered as an unweighted graph

(2) k-nearest neighbor graphs

Connect vertex v_i with vertex v_j if v_j is among the k-nearest neighbors of v_i

(this definition leads to a directed graph, as the neighborhood relationship is not symmetric)

There are two ways of making this graph undirected:

- k-nearest neighbor graph
- mutual k-nearest neighbor graph

(3) fully connected graph

Connect all points with positive similarity with each other

Weight all edges by s_{ij}

As the graph should represent the local neighborhood relationships, this construction is only useful if the similarity function itself models local neighborhoods

An example for such a similarity function is the Gaussian similarity function

$$s(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}$$

where the parameter σ controls the width of the neighborhoods and plays a similar role as the parameter ϵ in case of the ϵ-neighborhood graph

---

All these 3 similarity graphs are used in spectral clustering.

There are no theoretical results on the question of how the choice of the similarity graph influences the spectral clustering result.

Graph Laplacians

- main tools for spectral clustering: graph Laplacian matrices
- these are studied in the field of spectral graph theory

  Chung, Fan R. K. Spectral graph theory. CBMS Regional Conference Series in Mathematics, volume 92. AMS 1997. xii+207 pp.

  keywords: Cheeger constants and diameters, log-Sobolev constants, Harnack inequalities
- in the literature there is no unique convention for exactly which matrix is called "graph Laplacian"
- assume that G is an undirected, weighted graph with weight matrix W, s.t. $w_{ij} = w_{ji} \geq 0$
- not necessarily assume that eigenvectors of a matrix are normalized
- example: the constant vector $\mathbb{1}$ and a multiple $\alpha\mathbb{1}$ for some $\alpha > 0$, will be considered as the same eigenvectors
- Eigenvalues will always be ordered increasingly, respecting multiplicities.
  "the first k eigenvectors": the eigenvectors corresponding to the k smallest eigenvalues.

Un-normalized graph Laplacian

Definition

$$L = D - W$$

Properties

- For every vector $f \in \mathbb{R}^n$ we have: $f^T \cdot L \cdot f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$
- L is symmetric and positive semi-definite
- The smallest eigenvalue of L is 0, the corresponding eigenvector is $\mathbb{1}$
- L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

The unnormalized graph Laplacian and its eigenvalues and eigenvectors can be used to describe many properties of graphs, for example here is an important result for spectral clustering:

Proposition (Number of connected components and the spectrum of L)

Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

Normalized graph Laplacians

Definition

$$L_{sym} = D^{-1/2} \cdot L \cdot D^{-1/2} \qquad \text{it is a symmetric matrix}$$

$$L_{rw} = D^{-1} \cdot L \qquad \text{closely related to a random walk}$$

Properties

- For every vector $f \in \mathbb{R}^n$ we have: $f^T \cdot L_{sym} \cdot f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$
- λ is an eigenvalue of L_{rw} with eigenvector u IFF λ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2}u$
- λ is an eigenvalue of L_{rw} with eigenvector u IFF λ and u solve the generalized eigenproblem $Lu = \lambda Du$
- 0 is an eigenvalue of L_{rw} with eigenvector $\mathbb{1}$
- 0 is an eigenvalue of L_{sym} with eigenvector $D^{1/2}\mathbb{1}$
- L_{sym} and L_{rw} are positive semi-definite and have n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Proposition (Number of connected components and the spectra of L_{sym}, L_{rw})

Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of both L_{sym} and L_{rw} equals the number of connected components $A_1, \ldots, A_k$ in the graph.

For L_{rw}, the eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

For L_{sym}, the eigenspace of eigenvalue 0 is spanned by the vectors $D^{1/2}\mathbb{1}_{A_1}, \ldots, D^{1/2}\mathbb{1}_{A_k}$ of those components.

Spectral Clustering Algorithms

general setup:

- assume that our data consists of n "points" $x_1, \ldots, x_n$ which can be arbitrary objects
- measure their pairwise similarities $s_{ij} = s(x_i, x_j)$ by some similarity function which is symmetric and non-negative
- denote the corresponding similarity $n \times n$ matrix by $S = (s_{ij})$

Unnormalized spectral clustering

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, # k of clusters to construct

1. Construct a similarity graph by one of the ways described previously. Let W be its weighted adjacency matrix
2. Compute the un-normalized Laplacian L
3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of L
4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns
5. For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U
6. Cluster the n points y_i in $\mathbb{R}^k$ with the k-means algorithm, into clusters $C_1, \ldots, C_k$

Output: clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$

Normalized spectral clustering

There are two different versions of normalized spectral clustering, depending which of the normalized graph Laplacians is used.

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, # k of clusters to construct

1. Construct a similarity graph by one of the ways described previously. Let W be its weighted adjacency matrix
2. Compute the un-normalized Laplacian L
3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$
4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns
5. For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U
6. Cluster the n points y_i in $\mathbb{R}^k$ with the k-means algorithm, into clusters $C_1, \ldots, C_k$

Output: clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$

Note that this algorithm uses the generalized eigenvectors of L, which according to a previous proposition correspond to the eigenvectors of the matrix L_{rw}. So in fact, the algorithm works with eigenvectors of the normalized Laplacian L_{rw}, and hence is called normalized spectral clustering.

Normalized spectral clustering (Ng,Jordan,Weiss, 2002)

The following normalized spectral clustering algorithm also uses a normalized Laplacian, but this time it is the matrix L_{sym} instead of L_{rw}

This algorithm needs to introduce an additional row normalization step which is not needed in the other algorithms

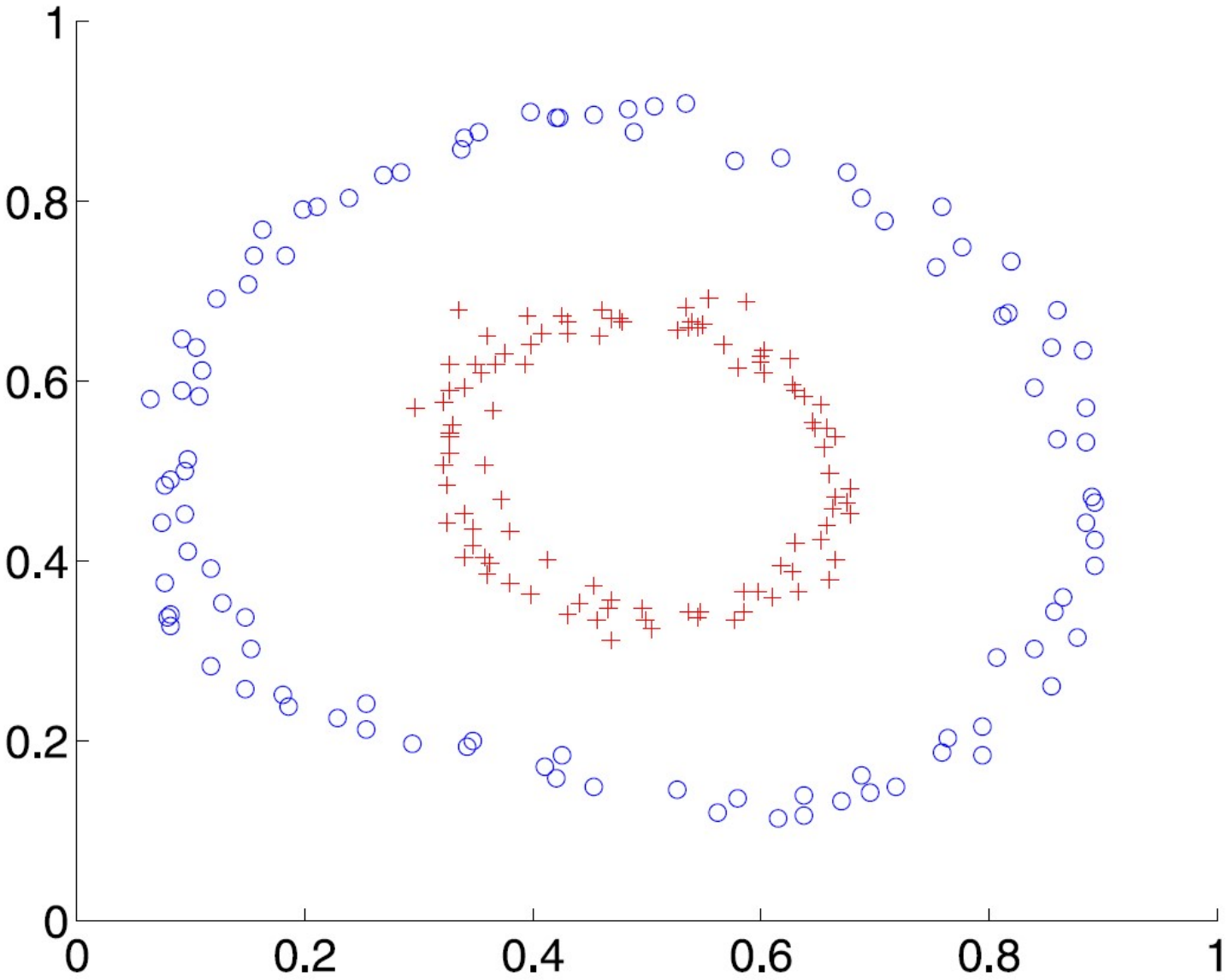
> Andrew Ng, Micheal Jordan, Yair Weiss, (2002). On spectral clustering: analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14 (pp. 849 - 856). MIT Press.

Cited by 9035 Source: Google Scholar

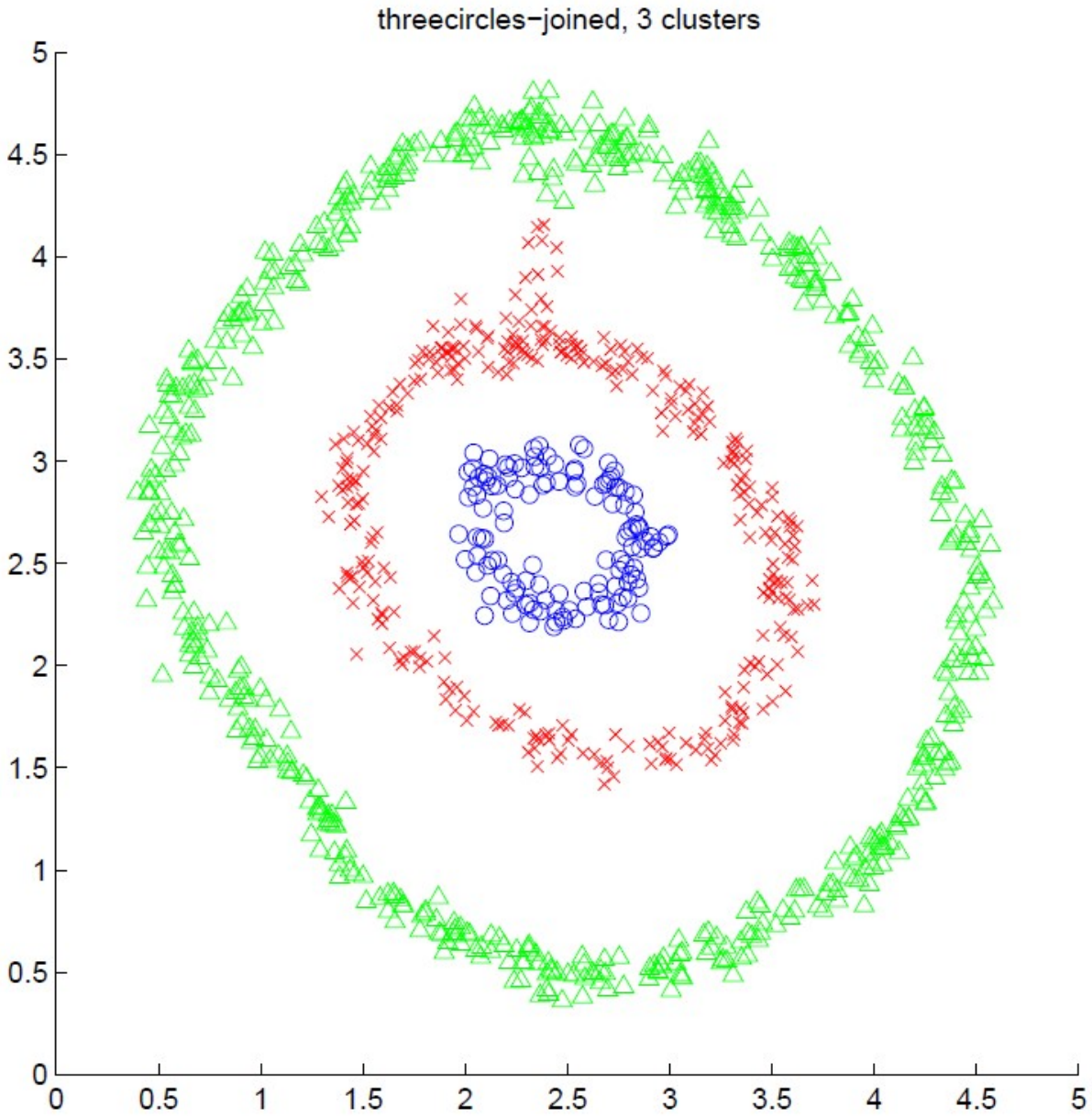
Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, # k of clusters to construct

1. Construct a similarity graph by one of the ways described previously. Let W be its weighted adjacency matrix
2. Compute the normalized Laplacian L_{sym}
3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of L_{sym}
4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns
5. Form the matrix $T \in \mathbb{R}^{n \times k}$ from U by normalizing rows to norm 1, that is set $t_{ij} = u_{ij} / \sqrt{\sum_k u_{ik}^2}$
6. For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T
7. Cluster the n points y_i in $\mathbb{R}^k$ with the k-means algorithm, into clusters $C_1, \ldots, C_k$

Output: clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$



TwoCircles data set with correct clustering


threecircles−joined, 3 clusters
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Perturbation approach & eigengap heuristic

- Perturbation theory studies the question of how eigenvalues and eigenvectors of a matrix A change if we add a small perturbation H, that is we consider the perturbed matrix $\tilde{A} = A + H$.

- Most perturbation theorems state that a certain distance between eigenvalues or eigenvectors of A and $\tilde{A}$ is bounded by a constant times a norm of H.

- The constant usually depends on which eigenvalue we are looking at, and how far this eigenvalue is separated from the rest of the spectrum.

- The justification of spectral clustering is then the following: Let us first consider the "ideal case" where the between-cluster similarity is exactly 0. Then the first k eigenvectors of L or L_{rw} are the indicator vectors of the clusters. In this case, the points $y_i \in \mathbb{R}^k$ constructed in the spectral clustering algorithms have the form $(0, \ldots, 0, 1, 0, \ldots, 0)$ where the position of the 1 indicates the connected component this point belongs to. In particular, all y_i belonging to the same connected component coincide. The k-means algorithm will trivially find the correct partition by placing a center point on each of the points $(0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{R}^k$.

- In a "nearly ideal case" where we still have distinct clusters, but the between-cluster similarity is not exactly 0, we consider the Laplacian matrices to be perturbed versions of the ones of the ideal case. Perturbation theory then tells us that the eigenvectors will be very close to the ideal indicator vectors. The points y_i might not completely coincide with $(0, \ldots, 0, 1, 0, \ldots, 0)$, but do so up to some small error term. Hence, if the perturbations are not too large, then k-means algorithm will still separate the groups from each other.

- The formal basis for the perturbation approach to spectral clustering is the Davis-Kahan theorem from matrix perturbation theory. It bounds the difference between eigenspaces of symmetric matrices under perturbations.

- Distances between subspaces are usually measured using "canonical angles" (also called "principal angles").

- Let V_1 and V_2 be two p-dimensional subspaces of $\mathbb{R}^d$, and V_1 and V_2 two matrices such that their columns form orthonormal systems for V_1 and V_2, respectively. Then the cosines $\cos \Theta_i$ of the principal angles Θ_i are the singular values of $V_1^t V_2$. The matrix $\sin \Theta(V_1, V_2)$ denotes the diagonal matrix with the sine of the canonical angles on the diagonal.

Davis-Kahan Theorem

Let $A, H \in \mathbb{R}^{n\times n}$ be symmetric matrices, $||\cdot||$ be the Frobenius norm (or the 2-norm), $\tilde{A} := A + H$ a perturbed version of A. Let $S_1 \subset \mathbb{R}$ be an interval. Denote by $\sigma_{S_1}(A)$ the set of eigenvalues of A which are contained in S_1, and by V_1 the eigenspace corresponding to all those eigenvalues. Denote by $\sigma_{S_1}(\tilde{A})$ and $\tilde{V_1}$ the analogous quantities for $\tilde{A}$. Define the distance between S_1 and the spectrum of A outside of S_1 as

$$\delta = \min\{|\lambda - s|; \lambda \text{ eigenvalue of } A, \lambda \notin S_1, s \in S_1\}.$$

Then the distance $d(V_1, \tilde{V_1}) := \sin\Theta(V_1, \tilde{V_1})$ between the two subspaces V_1, $\tilde{V_1}$ is bounded by

$$d(V_1, \tilde{V_1}) \leq \frac{||H||}{\delta}$$

Davis-Kahan theorem for unnormalized Laplacian

- The matrix A will correspond to the graph Laplacian L in the ideal case where the graph has k connected components.
- The matrix $\tilde{A}$ corresponds to a perturbed case, where due to noise the k components in the graph are no longer completely disconnected, but they are only connected by few edges with low weight.
- Denote the corresponding graph Laplacian of this case by $\tilde{L}$.
- For spectral clustering we need to consider the first k eigenvalues and eigenvectors of $\tilde{L}$.
- Denote the eigenvalues of L by $\lambda_1, \ldots, \lambda_n$
- Denote the eigenvalues of the perturbed Laplacian $\tilde{L}$ by $\tilde{\lambda_1}, \ldots, \tilde{\lambda_n}$.
- Crucial step: Choosing the interval S_1 so that both the first k eigenvalues of $\tilde{L}$ and the first k eigenvalues of L are contained in it.

- This is easier, the smaller the perturbation $H = L - \tilde{L}$ and the larger the eigengap $|\lambda_k - \lambda_{k+1}|$ is.

- If we manage to find such a set, then the Davis-Kahan theorem tells us that the eigenspaces corresponding to the first k eigenvalues of the ideal matrix L and the first k eigenvalues of the perturbed matrix $\tilde{L}$ are very close to each other, that is their distance is bounded by $||H||/\delta$.

- Then, as the eigenvectors in the ideal case are piecewise constant on the connected components, the same will approximately be true in the perturbed case. How good "approximately" is depends on the norm of the perturbation $||H||$ and the distance δ between S_1 and the $(k+1)$st eigenvector of L. If the set S_1 has been chosen as the interval $[0, \lambda_k]$, then δ coincides with the spectral gap $|\lambda_k - \lambda_{k+1}|$.

- We can see from the theorem that the larger this eigengap is, the closer the eigenvectors of the ideal case and the perturbed case are, and hence the better spectral clustering works.

- The size of the eigengap can also be used in a different context as a quality criterion for spectral clustering, namely when choosing the number k of clusters to construct.

- If the perturbation H is too large or the eigengap is too small, we might not find a set S_1 such that both the first k eigenvalues of L and $\tilde{L}$ are contained in S_1.

- In this case, we need to make a compromise by choosing the set S_1 to contain the first k eigenvalues of L, but maybe a few more or less eigenvalues of $\tilde{L}$.

- The statement of the theorem then becomes weaker in the sense that:
  (1) either we do not compare the eigenspaces corresponding to the first k eigenvectors of L and $\tilde{L}$, but the eigenspaces corresponding to the first k eigenvectors of L and the first $\tilde{k}$ eigenvectors of $\tilde{L}$
  (where $\tilde{k}$ is the number of eigenvalues of $\tilde{L}$ contained in S_1).
  (2) Or, it can happen that δ becomes so small that the bound on the distance between $d(V_1, \tilde{V_1})$ blows up so much that it becomes useless.

Some other important papers

- Spectral methods have become increasingly popular for clustering. These algorithms cluster data given in the form of a graph. One spectral approach to semi-supervised clustering is the SPECTRAL-LEARNING algorithm:

  `Sepandar D. Kamvar, Dan Klein, Christopher D. Manning:`
  `Spectral Learning. IJCAI 2003, pp. 561-566`

- Semi-supervised clustering algorithms aim to improve clustering results using limited supervision. The supervision is generally given as pairwise constraints; such constraints are natural for graphs, yet most semi-supervised clustering algorithms are designed for data represented as vectors. Unification of vector-based and graph-based approaches:

  `Semi-supervised graph clustering: a kernel approach`
  `Brian Kulis, Sugato Basu, Inderjit Dhillon, Raymond Mooney`
  `Machine Learning, volume 74, pp. 1-22, 2009`