Spectral clustering of graphs

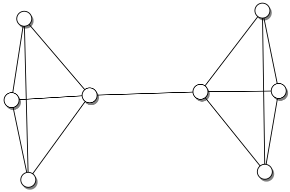
2016 CRM Summer School

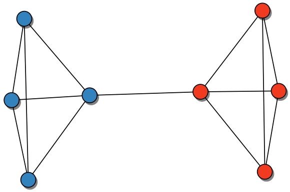
Spectral Theory and Applications

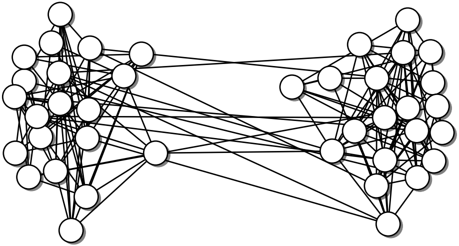
J.-G. Young

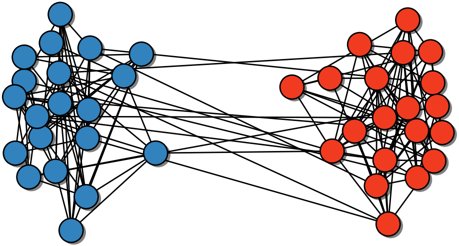
July 13, 2016

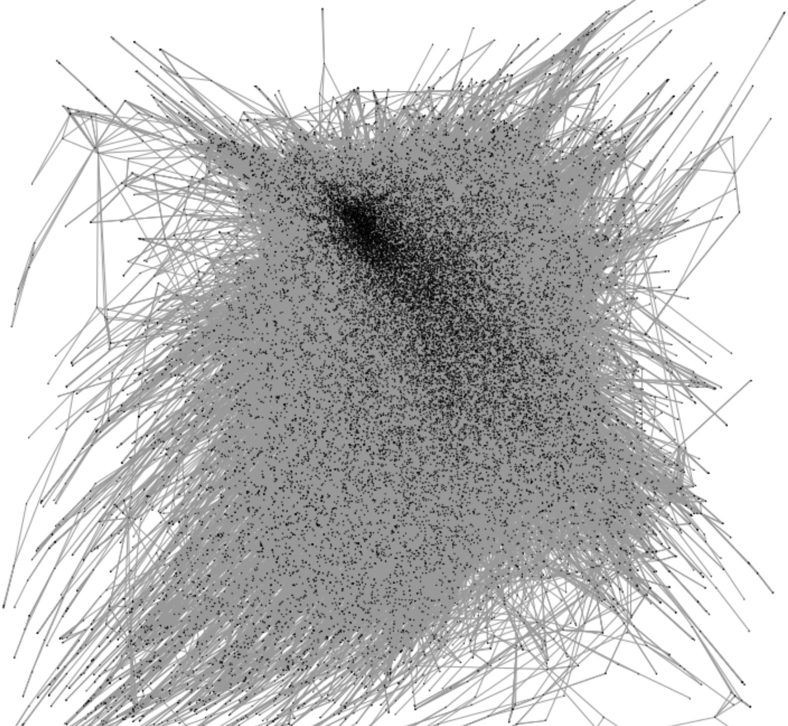
Département de physique, de génie physique, et d'optique
Université Laval, Québec, Canada

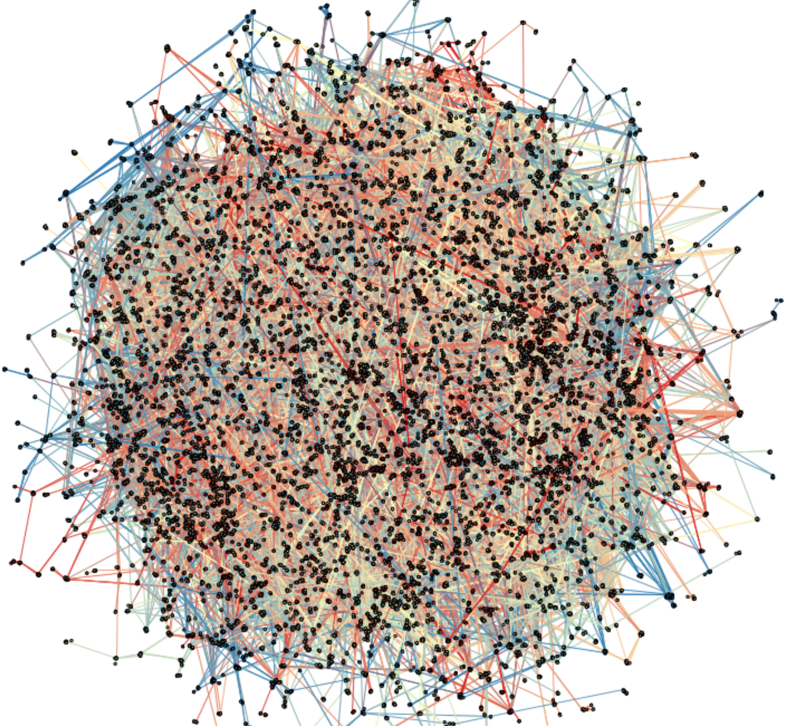












Content

Motivations

Graph Clustering

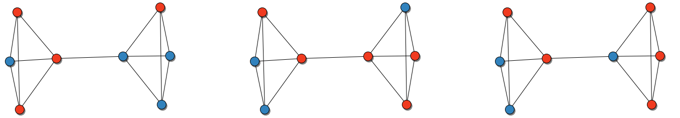
Formal definition

We have the vertex set $V(G)$ of an undirected graph $G(V, E)$.

We want to identify the partition $\mathscr{B}(V)$ of $V(G)$ which optimizes an *objective function*

$$f : \mathscr{B}, G \to \mathbb{R}$$

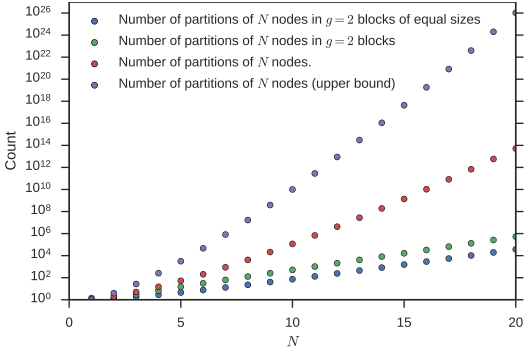
over the set of all partitions $\mathscr{B}(V)$.



Search space of graph clustering

Major hurdle :

EXPONENTIAL dependency of the number of solutions in $N = |V|$.



This is true even if the search space is heavily constrained.

Hardness of clustering

(constrained) graph clustering is in NP-Hard

- P : Problems solvable in polynomial time (*easy*)
- NP : Problems solvable in non-deterministic polynomial time (*hard*)
- NP-Complete : Equivalence class in NP (*hard*)
- NP-Hard : Problems which are at least as hard as the hardest problem in NP-Complete (*hardest*)

Bisection : the spectral method

Matrix formulation of graph bisection (1 of 2)

We will consider objective functions of the form

$$f(\{\sigma_i\}, G) = \frac{1}{2}\sum_{ij} h_{ij}^{(in)}(G)\delta_{\sigma_i\sigma_j} + \sum_{ij} h_{ij}^{(out)}(G)\bar{\delta}_{\sigma_i\sigma_j}$$
$$= \frac{1}{2}\sum_{ij} [h_{ij}^{(in)}(G) - h_{ij}^{(out)}(G)]\delta_{\sigma_i\sigma_j} \ .$$

Definitions

δ_{ij} : Kronecker delta.
σ_i : index of the block of vertex $v_i \in V$. Precisely $\sigma_i = r \implies v_i \in B_r$.
$h_{ij}^{(in)}$: cost associated to putting v_i, v_j in the same block.
$h_{ij}^{(out)}$: cost associated to putting v_i, v_j in different blocks.

Matrix formulation of graph bisection (2 of 2)

In GRAPH BISECTION, either $v_i \in B_1$ or $v_i \in B_2$.

We denote this with

$$s_i s_j = \begin{cases} 1 & \text{if } \sigma_i \neq \sigma_j \\ -1 & \text{otherwise} \end{cases} .$$

Matrix formulation of graph bisection (2 of 2)

In GRAPH BISECTION, either $v_i \in B_1$ or $v_i \in B_2$.

We denote this with

$$s_i s_j = \begin{cases} 1 & \text{if } \sigma_i \neq \sigma_j \\ -1 & \text{otherwise} \end{cases} .$$

Defining the *indicator vector* $\boldsymbol{s}$ and *objective matrix* $\boldsymbol{H}$, we rewrite the objective function as the QUADRATIC FORM

$$f(\{\sigma_i\}, G) \equiv \frac{1}{4}\boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s} + C .$$

Example of a objective matrix (1 of 5)

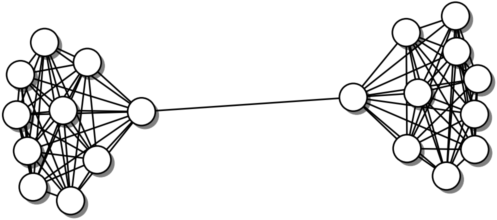
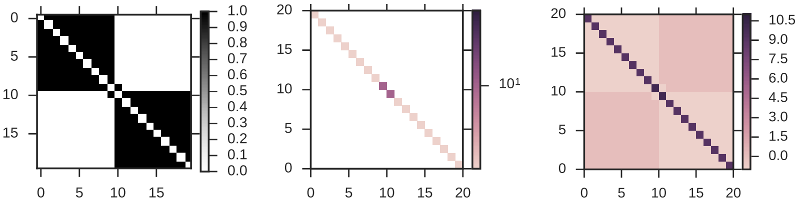


FIGURE – Barbell graph $B(n_1, n_2)$ with $n_1 = 10$, $n_2 = 0$

Combinatorial Laplacian

- A is the adjacency matrix of G
- D is the diagonal matrix of the degrees $k_i = \sum_{j=1}^{N} a_{ij}$
- $L := D - A$.



Example of a objective matrix (3 of 5)

The combinatorial Laplacian *counts* the number of edges between blocks.

$$f_{\text{Lap}} = \frac{1}{4}\boldsymbol{s}^T\boldsymbol{L}\boldsymbol{s} = \frac{1}{4}\boldsymbol{s}^T\boldsymbol{D}\boldsymbol{s} - \frac{1}{4}\boldsymbol{s}^T\boldsymbol{A}\boldsymbol{s}$$

Define $m(B_1, B_2)$ as the number of edges between blocks B_1, B_2 :

$$\frac{1}{4}\boldsymbol{s}^T\boldsymbol{D}\boldsymbol{s} = \frac{1}{4}\sum_{i=1}^{N} k_i s_i^2 = \frac{m}{2} \,,$$

$$\frac{1}{4}\boldsymbol{s}^T\boldsymbol{A}\boldsymbol{s} = \frac{m(B_1, B_1) + m(B_2, B_2) - m(B_1, B_2)}{2} \,.$$

Example of a objective matrix (4 of 5)

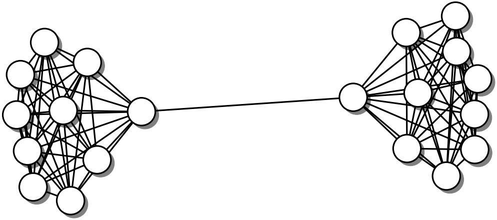


FIGURE – Barbell graph $B(n_1, n_2)$ with $n_1 = 10$, $n_2 = 0$

Example of a objective matrix (4 of 5)

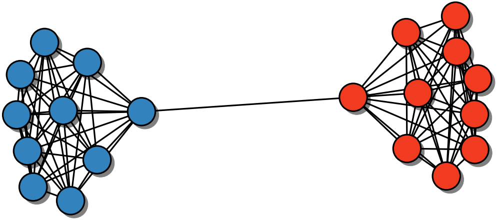


FIGURE – Barbell graph $B(n_1, n_2)$ with $n_1 = 10$, $n_2 = 0$

Example of a objective matrix (5 of 5)

Other standard objective matrix

- Adjacency matrix A
- Normalized Laplacians $L_{sym} = D^{-1/2} L D^{-1/2}$, $L_{rw} = D^{-1} L D$
- Modularity $Q = A - \langle A \rangle_{H_0}$

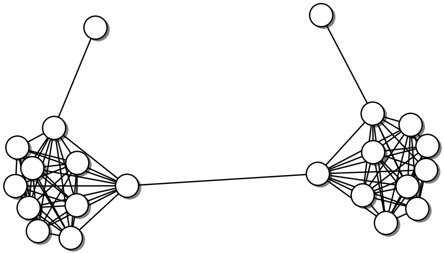


Figure – Modified Barbell graph.

Constrained bisection (2 of 4)

Unconstrained bisection

Optimize $f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s}$ subject to $\boldsymbol{s} \in \{-1, 1\}^N$.

Balanced partitions are often desirable. Unconstrained bisection does not ask for balance.

Constrained bisection (2 of 4)

Unconstrained bisection

$$\text{Optimize } f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s} \text{ subject to } \boldsymbol{s} \in \{-1, 1\}^N .$$

Balanced partitions are often desirable. Unconstrained bisection does not ask for balance.

$\exists$ two methods to constrains $\mathscr{B} = \{B_1, B_2\}$:

1. Modify f.
2. Reject bad solutions explicitly.

Constrained bisection (3 of 4)

Option 1 : Modifications to f

$$\widetilde{f}_{\text{Lap}} := \frac{f_{\text{Lap}}}{|B_1||B_2|} \,,$$

$$\bar{f}_{\text{Lap}} := \frac{f_{\text{Lap}}}{\text{vol}(B_1)\text{vol}(B_2)} \,.$$

Option 2 : Explicit constraint

$$\text{Optimize } f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s}$$

$$\text{subject to } \boldsymbol{s} \in \{-1, 1\}^N \text{ and } \boldsymbol{s}^T \mathbf{1} \leq \epsilon, \text{ with } \epsilon \geq 0 \,.$$

Constrained bisection (4 of 4)

Unconstrained bisection (*easy*)

Optimize $f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s}$ subject to $\boldsymbol{s} \in \{-1, 1\}^N$.

Constrained bisection (4 of 4)

Unconstrained bisection (*easy*)

$$\text{Optimize } f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s} \text{ subject to } \boldsymbol{s} \in \{-1, 1\}^N \ .$$

Constrained bisection (**hard**)

$$\text{Optimize } f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s}$$
$$\text{subject to } \boldsymbol{s} \in \{-1, 1\}^N \text{ and } \boldsymbol{s}^T \mathbf{1} \leq \epsilon, \text{ with } \epsilon \geq 0 \ .$$

Spectral algorithm for graph bisection (1 of 4)

Constrained bisection

$$\text{Optimize } f(\{\sigma_i\}, G) = \boldsymbol{s}^T \boldsymbol{H} \boldsymbol{s}$$

$$\text{subject to } \boldsymbol{s} \in \{-1, 1\}^N \text{ and } \boldsymbol{s}^T \mathbf{1} \leq \epsilon, \text{ with } \epsilon \geq 0 \; .$$

Dropping constraints turns bisection into an easy problem

Spectral algorithm for graph bisection (1 of 4)

Constrained bisection

$$\text{Optimize } f(\{\sigma_i\}, G) = \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x}$$

$$\text{SUBJECT TO } \boldsymbol{x} \in \mathbb{R}^N \text{ and } \boldsymbol{x}^T \mathbf{1} \leq \epsilon, \text{ with } \epsilon \geq 0 \,.$$

Dropping constraints turns bisection into an easy problem.

Spectral algorithm for graph bisection (2 of 4)

Justification : Suppose that $\boldsymbol{x} \in \mathbb{R}^N$ is a normalized eigenvector of $\boldsymbol{H}$ with eigenvalue λ_i. Then

$$f = \boldsymbol{x}_i^T \boldsymbol{H} \boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i^T \boldsymbol{x}_i = \lambda_i$$

If we have ordered eigenvectors (accounting for multiplicities),

$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$$

$\implies$ the optima of f correspond to extremal eigenvalues.

Spectral algorithm for graph bisection (3 of 4)

The continuous optimization perspective

$$f = \boldsymbol{x}_i^T \boldsymbol{H} \boldsymbol{x}_i = \sum_{ij} h_{ij} x_i x_j$$

OPTIMA of f are found by setting $\{\partial_{x_i}[f]\}$ to zero.

We avoid trivial solutions $x_i = 0\ \forall i$, by asking $\sum_i x_i^2 = \Delta, \Delta > 0$

$$\frac{\partial}{\partial x_r}\left[\sum_{ij} h_{ij} x_i x_j - \lambda\left(\sum_i x_i - \Delta\right)\right] = 0 \qquad (\Delta > 0)$$

Using $\partial_{x_r}[x_i] = \delta_{ir}$, we find that

$$\sum_j H_{ij} x_j = \lambda x_i \quad \Leftrightarrow \quad \boldsymbol{H}\boldsymbol{x} = \lambda \boldsymbol{x}$$

Spectral algorithm for graph bisection (4 of 4)

We have relaxed $\boldsymbol{s} \to \boldsymbol{x}$.

How do we recover $\boldsymbol{s}$?

Spectral algorithm for graph bisection (4 of 4)

We have relaxed $\boldsymbol{s} \to \boldsymbol{x}$.
How do we recover $\boldsymbol{s}$?

- In the BISECTION, we can show that the *sign* of $x_i \in \boldsymbol{x}$ is a good predictor of the nearest $\boldsymbol{s}$.

Spectral algorithm for graph bisection (4 of 4)

We have relaxed $\boldsymbol{s} \to \boldsymbol{x}$.
How do we recover $\boldsymbol{s}$?

- In the BISECTION, we can show that the *sign* of $x_i \in \boldsymbol{x}$ is a good predictor of the nearest $\boldsymbol{s}$.
- In general, we can use K-Means to minimize

$$\operatorname{argmin}_{\mathscr{B}} \sum_{r=1}^{g} \sum_{i \in B_r} ||\boldsymbol{x}_i - \boldsymbol{\mu}_r||^2$$

IMPORTANT CAVEAT : Reject solutions that do not satisfy $\boldsymbol{x}^T \mathbf{1} \leq \epsilon$.

Concrete examples (1 of 3)

Modified Barbell graph

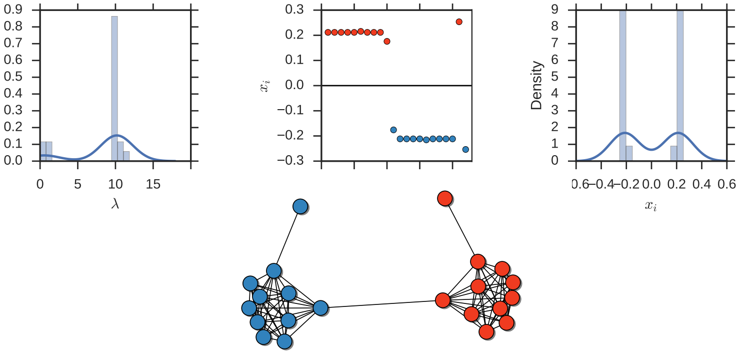


Figure – (*left*) Density of the eigenvalues of $\boldsymbol{L}$. (*middle*) Values of the elements of $\boldsymbol{x}_2$ (*right*) Distribution of the elements of $\boldsymbol{x}_2$ in $\mathbb{R}^1$.

PLANTED PARTITION GRAPH (EASY CASE)

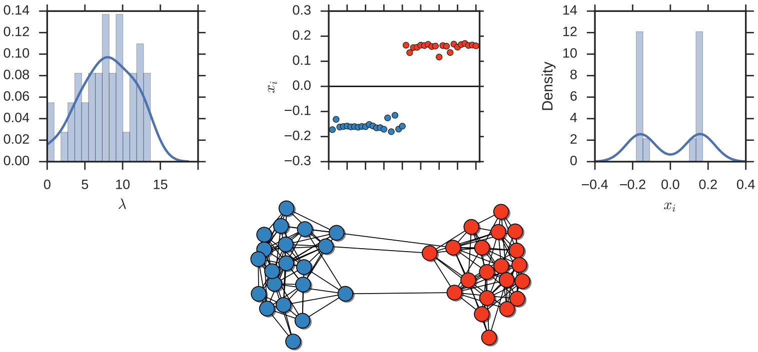


FIGURE – (*left*) Density of the eigenvalues of $\boldsymbol{L}$. (*middle*) Values of the elements of $\boldsymbol{x}_2$ (*right*) Distribution of the elements of $\boldsymbol{x}_2$ in $\mathbb{R}^1$.

Concrete examples (3 of 3)

Planted partition graph (hard case)

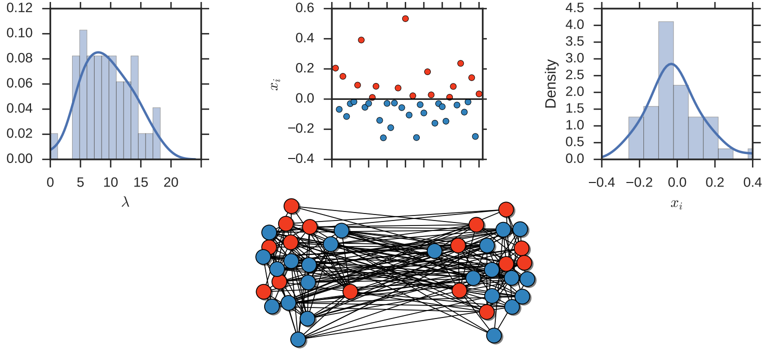


Figure – (*left*) Density of the eigenvalues of $\boldsymbol{L}$. (*middle*) Values of the elements of $\boldsymbol{x}_2$ (*right*) Distribution of the elements of $\boldsymbol{x}_2$ in $\mathbb{R}^1$.

General case : graph clustering

Matrix formulation of graph clustering (1 of 2)

Recall that we optimize objective functions of the form

$$f(\{\sigma_i\}, G) = \frac{1}{2}\sum_{ij}[h_{ij}^{(in)}(G) - h_{ij}^{(out)}(G)]\delta_{\sigma_i\sigma_j}\ .$$

If the partition has $g \geq 2$ block $\mathscr{B}$, we must use indicator VECTORS to represent $\delta_{\sigma_i\sigma_j}$.

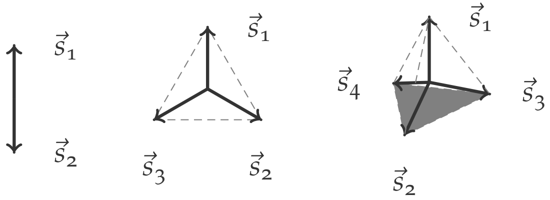


Figure – Corners of regular $(g-1)$-simplices.

Matrix formulation of graph clustering (2 of 2)

The indicator vectors satisfy

$$\boldsymbol{s}_i^T \boldsymbol{s}_i = \begin{cases} 1 - \frac{1}{g} & \text{if } \sigma_i \neq \sigma_j \\ -\frac{1}{g} & \text{otherwise} \end{cases} \quad . \tag{1}$$

$$\begin{aligned} f(\{\sigma_i\}, G) &= \frac{1}{2} \sum_{ij} [h_{ij}^{(in)}(G) - h_{ij}^{(out)}(G)] \delta_{\sigma_i \sigma_j} \\ &= \frac{1}{2} \mathrm{Tr}\big(\boldsymbol{S}^T \boldsymbol{H} \boldsymbol{S}\big) + C \end{aligned}$$

$\boldsymbol{S}$ is the $N \times g - 1$ matrix with vector $\boldsymbol{s}_i$ on row i.

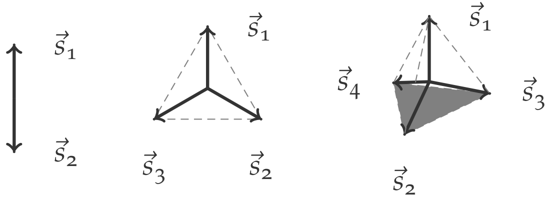
Constrained graph clustering

$$\text{Optimize } f(\{\sigma_i\}, G) = \mathrm{Tr}(\boldsymbol{S}^T \boldsymbol{H} \boldsymbol{s})$$

$$\text{subject to } \boldsymbol{S} \in (g-1)\text{-dimensional simplex}$$

and

$$\boldsymbol{S}^T \mathbf{1} \leq \mathbf{1}\epsilon, \text{ with } \epsilon \geq 0\ .$$



Optimal solutions

Suppose that $\boldsymbol{X}$ is a matrix of normalized eigenvectors of $\boldsymbol{H}$ such that

$$\boldsymbol{H}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\Lambda}$$

$\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues.

We see

$$f = \frac{1}{2}\mathrm{Tr}\big(\boldsymbol{X}^T\boldsymbol{H}\boldsymbol{X}\big) = \frac{1}{2}\mathrm{Tr}\big(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\Lambda}\big) = \frac{1}{2}\sum_{i=1}^{g-1}\lambda_i$$

$\Longrightarrow$ the optima of f are given by sums of extremal eigenvalues.

Continuous optimization perspective

OPTIMA of f are found by setting $\{\partial_{X_{rs}}[f]\}$ to zero.

$$f = \frac{1}{2}\mathrm{Tr}\big(\boldsymbol{X}^T\boldsymbol{H}\boldsymbol{X}\big)$$

We avoid trivial solutions $X_{rs} = 0\ \forall i$, by asking $\boldsymbol{X}^T\boldsymbol{X} = \Delta\boldsymbol{I}$

$$\frac{\partial}{\partial\boldsymbol{X}}\left[\frac{1}{2}\mathrm{Tr}\big(\boldsymbol{X}^T\boldsymbol{H}\boldsymbol{X}\big) - \mathrm{Tr}\big(\boldsymbol{X}(\boldsymbol{\Lambda} + \Delta\boldsymbol{I})\boldsymbol{X}^T\big)\right] = 0 \qquad (\Delta > 0)$$

This leads to

$$\boldsymbol{H}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\Lambda}$$

Because we have the identities

$$\frac{\partial}{\partial\boldsymbol{X}}\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}) = (\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{X}$$
$$\frac{\partial}{\partial\boldsymbol{X}}\mathrm{Tr}(\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^T) = \boldsymbol{X}\big(\boldsymbol{A} + \boldsymbol{A}^T\big)$$

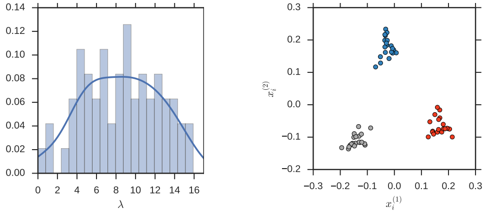
Spectral clustering algorithm

Input : number of blocks g, objective matrix $\boldsymbol{H}$, tolerance ϵ

1. Compute the g largest (smallest) eigenvalue of $\boldsymbol{H}$
2. Construct the $N \times (g-1)$ matrix of eigenvectors $\boldsymbol{X}$
3. Verify that $\boldsymbol{X}^T \mathbf{1} \leq \epsilon \mathbf{1}$ (element-wise).
   If not, replace the faulty eigenvector.
4. Cluster the elements of $\boldsymbol{X}$ in $\mathbb{R}^{g-1}$ with K-Means.

Return : The clusters in $\mathbb{R}^{g-1}$.

Planted partition graph (3 blocks case)



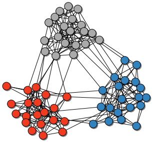


Figure – (*left*) Density of the eigenvalues of $\boldsymbol{L}$. (*right*) Elements of $\boldsymbol{x}_2$ versus the element of $\boldsymbol{x}_3$ in $\mathbb{R}^2$.

The number of clusters

The optimal nb. of blocks is predicted by the EIGENGAP

$$\Delta_i = \frac{(\lambda_{i+1} - \lambda_i)}{N}$$

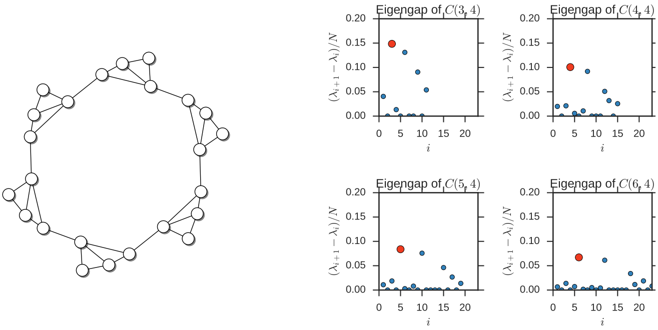


FIGURE – (*left*) Caveman graph $C(6,4)$ (*right*) Eigengap of $C(\ell,4)$, $\ell = 4,..,6$.

Two experiments

Zachary Karate Club

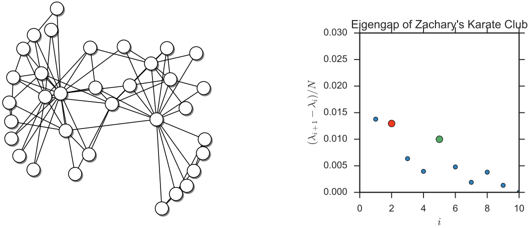
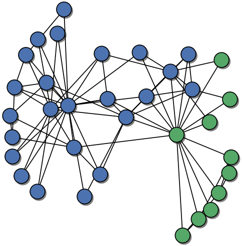


Figure – (*left*) Graph of interactions (*right*) Statistics of the eigengap [Laplacian matrix].

Zachary Karate Club



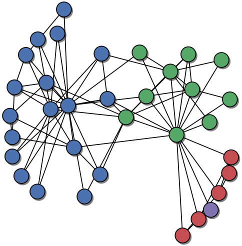
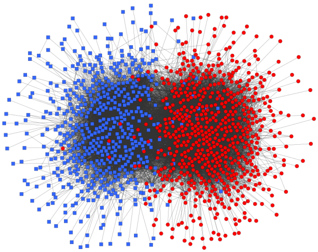
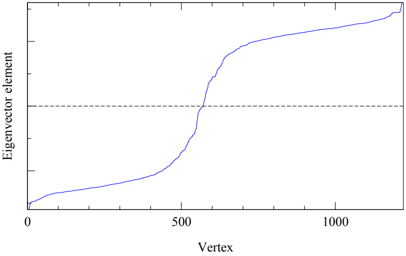


Figure – Bisection and 4-way clustering with the Laplacian matrix.

Political blogs





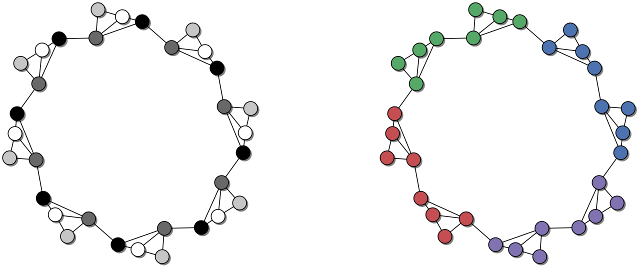
Dataset : L. A. Adamic and N. Glance, *The political blogosphere* (2005)

Figures : M.E.J. Newman, *Spectral methods for network community detection and graph partitioning*, (2013)

Conclusion

Take home message

- Constrained clustering is hard (NP-Hard)
- Relaxing the discrete constraint $\implies$ spectral algorithm
- The spectral approach arises from the continuous optimization perspective
- The framework is general, arbitrary $\boldsymbol{H}$.



Supplementary Material

The slides, lecture notes and `python` notebook are online at

`www.jgyoung.ca/crm2016/`

Recommended reading

- Excellent tutorial :

  U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and computing, 17 (2007), pp. 395–416.

- Clustering from first principles :

  M. A. Riolo and M. E. J. Newman, *First-principles multiway spectral partitioning of graphs*, J. Complex Netw., 2 (2014), pp. 121–140.