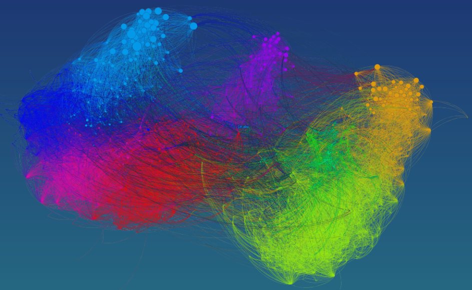
Graphs in Machine Learning

Michal Valko

Inria Lille - Nord Europe, France

Partially based on material by: Ulrike von Luxburg, Gary Miller, Doyle & Schnell, Daniel Spielman



Previous Lecture

- similarity graphs
  - different types
  - construction
  - sources of graphs
  - practical considerations
- spectral graph theory
- Laplacians and their properties
  - symmetric and asymmetric normalization
- random walks
- recommendation on a bipartite graph

This Lecture

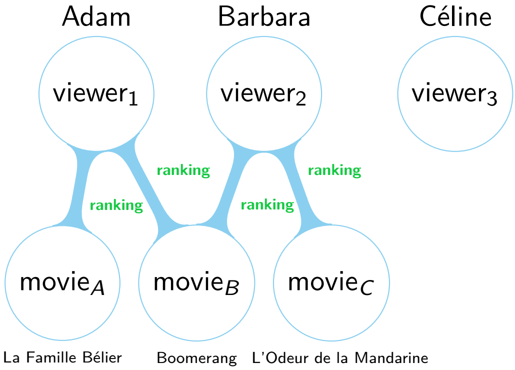
- resistive networks
  - recommendation score as a resistance?
  - Laplacian and resistive networks
  - resistance distance and random walks
- geometry of the data and the connectivity
- spectral clustering
- manifold learning with Laplacians

Next Class: Lab Session

- 19. 10. 2015 by Daniele.Calandriello@inria.fr
- Salle Condorcet
- Download the image and set it up BEFORE the class
- Matlab/Octave
- Short written report (graded)
- All homeworks together account for 40% of the final grade
- Content
  - Graph Construction
  - Test sensitivity to parameters: σ, k, ε
  - Spectral Clustering
  - Spectral Clustering vs. k-means
  - Image Segmentation

Use of Laplacians: Movie recommendation

How to do movie recommendation on a bipartite graph?



Question: *Do we recommend* <u>*L'Odeur de la Mandarine*</u> *to Adam?*

Let's compute some $\mathrm{score}(v, m)$!

Use of Laplacians: Movie recommendation

How to compute the $\mathrm{score}(v, m)$? Using some **graph distance**!

Idea$_1$: maximally weighted path

$$\mathrm{score}(v, m) = \max_{vPm} \mathrm{weight}(P) = \max_{vPm} \sum_{e \in P} \mathrm{ranking}(e)$$

Idea$_2$: change the path weight

$$\mathrm{score}_2(v, m) = \max_{vPm} \mathrm{weight}_2(P) = \max_{vPm} \min_{e \in P} \mathrm{ranking}(e)$$

Idea$_3$: consider everything

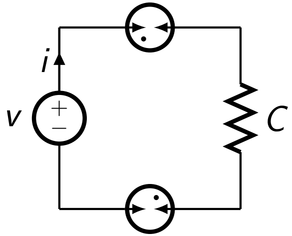
$$\mathrm{score}_3(v, m) = \text{max flow from } m \text{ to } v$$

Laplacians and Resistive Networks

How to compute the $\mathrm{score}(v, m)$?

Idea$_4$: view edges as conductors

$\mathrm{score}_4(v, m)$ = effective resistance between m and v



$C \equiv \mathrm{conductance}$

$R \equiv \mathrm{resistance}$

$i \equiv \mathrm{current}$

$V \equiv \mathrm{voltage}$

$$C = \frac{1}{R} \qquad i = CV = \frac{V}{R}$$

Resistive Networks

resistors in series

$$R = R_1 + \cdots + R_n \qquad C = \frac{1}{\frac{1}{C_1} + \cdots + \frac{1}{C_n}} \qquad i = \frac{V}{R}$$

conductors in parallel

$$C = C_1 + \cdots + C_n \qquad i = VC$$

Effective Resistance on a graph

Take two nodes: $a \neq b$. Let V_{ab} be the voltage between them and i_{ab} the current between them. Define $R_{ab} = \frac{V_{ab}}{i_{ab}}$ and $C_{ab} = \frac{1}{R_{ab}}$.

We treat the entire graph as a resistor!

Resistive Networks: Optional Homework (ungraded)

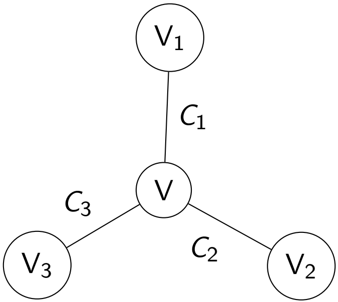
Show that R_{ab} is a metric space.

1. $R_{ab} \geq 0$
2. $R_{ab} = 0$ iff $a = b$
3. $R_{ab} = R_{ba}$
4. $R_{ac} \leq R_{ab} + R_{bc}$

The effective resistance is a distance!

How to compute effective resistance?

Kirchhoff's Law $\equiv$ flow in $=$ flow out



$V = \frac{C_1}{C} V_1 + \frac{C_2}{C} V_2 + \frac{C_3}{C} V_3$ (convex combination)

residual current $= CV - C_1 V_1 - C_2 V_2 - C_3 V_3$

Resistors: Where is the link with the Laplacian?

General case of the previous! $d_i = \sum_j c_{ij}$ = sum of conductances

$$\mathbf{L}_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -c_{ij} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

$\mathbf{v}$ = **voltage setting** of the nodes on graph.

$(\mathbf{Lv})_i$ = residual current at $\mathbf{v}_i$ — as we derived

Use: setting voltages and getting the current

Inverting $\equiv$ injecting current and getting the voltages

The net injected has to be zero - Kirchhoff's Law.

Resistors and the Laplacian: Finding R_{ab}

Let's calculate R_{1n} to get the **movie recommendation score**!

$$\mathbf{L}\begin{pmatrix} 0 \\ v_2 \\ \vdots \\ v_{n-1} \\ 1 \end{pmatrix} = \begin{pmatrix} i \\ 0 \\ \vdots \\ 0 \\ -i \end{pmatrix}$$

$$i = \frac{V}{R} \qquad V = 1 \qquad R = \frac{1}{i}$$

Return $R_{1n} = \dfrac{1}{i}$

Doyle and Snell: Random Walks and Electric Networks

https://math.dartmouth.edu/~doyle/docs/walks/walks.pdf

Resistors and the Laplacian: Finding R_{1n}

$\mathbf{Lv} = (i, 0, \ldots, -i)^\mathsf{T} \equiv$ **boundary valued problem**

For R_{1n}

- V_1 and V_n are the **boundary**
- $(v_1, v_2, \ldots, v_n)$ is **harmonic**
- $V_i \in$ **interior** (not boundary)
  - V_i is a **convex combination of its neighbors**

Resistors and the Laplacian: Finding R_{1n}

From the properties of electric networks (cf. Doyle and Snell) we inherit the useful properties of the Laplacians!

Example: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions (later in the course)

Maximum Principle

If $\mathbf{f}$ is harmonic then min and max are on the boundary.

Uniqueness Principle

If $\mathbf{f}$ and $\mathbf{g}$ are harmonic with the same boundary then $\mathbf{f} = \mathbf{g}$

Resistors and the Laplacian: Finding R_{1n}

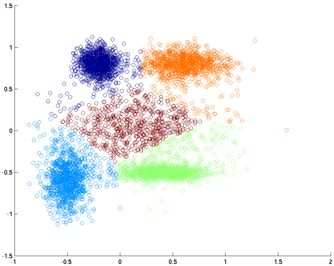
Alternative method to calculate R_{1n}:

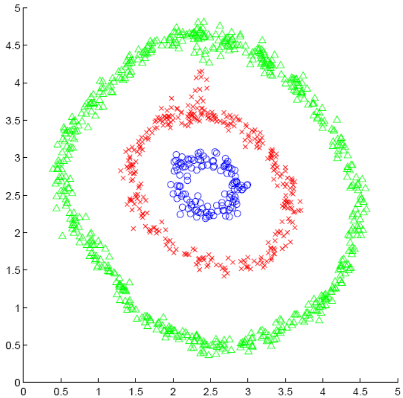
$$\mathbf{L}\mathbf{v} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \stackrel{\text{def}}{=} \mathbf{i}_{\text{ext}} \quad \text{Return} \quad R_{1n} = v_1 - v_n$$

Why?

Question: Does $\mathbf{v}$ exist? $\mathbf{L}$ does not have an inverse :(.
Solution: Instead of $\mathbf{v} = \mathbf{L}^{-1}\mathbf{i}_{\text{ext}}$ we take $\mathbf{v} = \mathbf{L}^{+}\mathbf{i}_{\text{ext}}$
Moore-Penrose pseudo-inverse solves LS
We get: $R_{1n} = v_1 - v_n = \mathbf{i}_{\text{ext}}^{\mathsf{T}}\mathbf{v} = \mathbf{i}_{\text{ext}}^{\mathsf{T}}\mathbf{L}^{+}\mathbf{i}_{\text{ext}}$.
Not unique: $\mathbf{1}$ in the nullspace of $\mathbf{L}$: $\mathbf{L}(\mathbf{v} + c\mathbf{1}) = \mathbf{L}\mathbf{v} + c\mathbf{L}\mathbf{1} = \mathbf{L}\mathbf{v}$

Application of Graphs for ML: **Clustering**

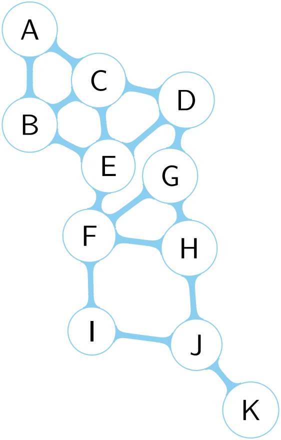




Application: Clustering - Recap

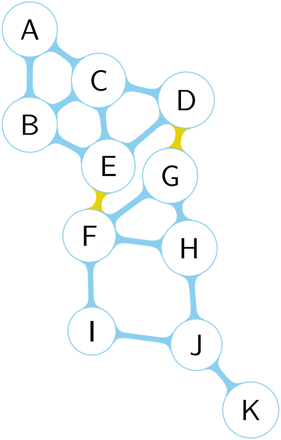
- What do we know about the **clustering** in general?
  - ill defined problem (different tasks $\rightarrow$ different paradigms)
  - inconsistent (wrt. Kleinberg's axioms)
  - number of clusters k need often be known
  - difficult to evaluate
- What do we know about **k-means**?
  - "hard" version of EM clustering
  - sensitive to initialization
  - optimizes for **compactness**
  - yet: algorithm-to-go

Spectral Clustering: Cuts on graphs



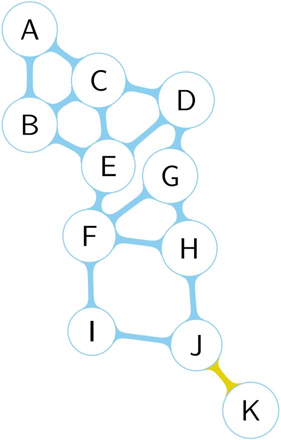
Defining the cut objective we get the clustering!

Spectral Clustering: Cuts on graphs



Defining the cut objective we get the clustering!

Spectral Clustering: Cuts on graphs



Defining the cut objective we get the clustering!

MinCut: $\mathrm{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$ Are we done?

Can be solved efficiently, but maybe not what we want ....

Spectral Clustering: Balanced Cuts

Let's balance the cuts!

MinCut

$$\mathrm{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

RatioCut

$$\mathrm{RatioCut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{|A|} + \frac{1}{|B|} \right)$$

Normalized Cut

$$\mathrm{NCut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right)$$

Spectral Clustering: Balanced Cuts

$$\mathrm{RatioCut}(A, B) = \mathrm{cut}(A, B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$$

$$\mathrm{NCut}(A, B) = \mathrm{cut}(A, B)\left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}\right)$$

Can we compute this? RatioCut and NCut are NP hard :(

Approximate!

Spectral Clustering: Relaxing Balanced Cuts

Relaxation for (simple) balanced cuts

$$\min_{A,B} \mathrm{cut}(A, B) \quad \text{s.t.} \quad |A| = |B|$$

Graph function $\mathbf{f}$ for cluster membership: $f_i = \begin{cases} 1 & \text{if } V_i \in A, \\ -1 & \text{if } V_i \in B. \end{cases}$

What it is the cut value with this definition?

$$\mathrm{cut}(A, B) = \sum_{i \in A, j \in B} w_{i,j} = \frac{1}{4} \sum_{i,j} w_{i,j} (f_i - f_j)^2 = \frac{1}{2} \mathbf{f}^\mathsf{T} \mathbf{L} \mathbf{f}$$

What is the relationship with the **smoothness** of a graph function?

Spectral Clustering: Relaxing Balanced Cuts

$$\mathrm{cut}(A,B) = \sum_{i\in A, j\in B} w_{i,j} = \frac{1}{4}\sum_{i,j} w_{i,j}(f_i - f_j)^2 = \frac{1}{2}\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$$

$|A| = |B| \implies \sum_i f_i = 0 \implies \mathbf{f} \perp \mathbf{1}_n$

$\|\mathbf{f}\| = \sqrt{n}$

objective function of spectral clustering

$$\min_{\mathbf{f}} \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} \quad \text{s.t.} \quad f_i = \pm 1, \quad \mathbf{f} \perp \mathbf{1}_n, \quad \|\mathbf{f}\| = \sqrt{n}$$

Still NP hard :($\rightarrow$

Relax even further!

$$\cancel{f_i = \pm 1} \quad \rightarrow \quad f_i \in \mathbb{R};$$

Spectral Clustering: Relaxing Balanced Cuts

objective function of spectral clustering

$$\min_{\mathbf{f}} \mathbf{f}^{\mathsf{T}}\mathbf{L}\mathbf{f} \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad \mathbf{f} \perp \mathbf{1}_n, \quad \|\mathbf{f}\| = \sqrt{n}$$

Rayleigh-Ritz Theorem

If $\lambda_1 \leq \cdots \leq \lambda_n$ are the eigenvectors of real symmetric $\mathbf{M}$ then

$$\lambda_1 = \min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}}{\mathbf{x}^{\mathsf{T}}\mathbf{x}} = \min_{\mathbf{x}^{\mathsf{T}}\mathbf{x}=1} \mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}$$

$$\lambda_n = \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}}{\mathbf{x}^{\mathsf{T}}\mathbf{x}} = \max_{\mathbf{x}^{\mathsf{T}}\mathbf{x}=1} \mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}$$

$\frac{\mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}}{\mathbf{x}^{\mathsf{T}}\mathbf{x}} \equiv$ **Rayleigh quotient**

How can we use it?

Spectral Clustering: Relaxing Balanced Cuts

objective function of spectral clustering

$$\min_{\mathbf{f}} \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad \mathbf{f} \perp \mathbf{1}_n, \quad \|\mathbf{f}\| = \sqrt{n}$$

Generalized Rayleigh-Ritz Theorem

If $\lambda_1 \leq \cdots \leq \lambda_n$ are the eigenvectors of real symmetric $\mathbf{M}$ and $\mathbf{v}_1, \ldots, \mathbf{v}_n$ the corresponding orthogonal eigenvalues, then for $k = 1 : n-1$

$$\lambda_{k+1} = \min_{\mathbf{x} \neq 0, \mathbf{x} \perp \mathbf{v}_1, \ldots \mathbf{v}_k} \frac{\mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}}{\mathbf{x}^\mathsf{T}\mathbf{x}} = \min_{\mathbf{x}^\mathsf{T}\mathbf{x}=1, \mathbf{x} \perp \mathbf{v}_1, \ldots \mathbf{v}_k} \mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}$$

$$\lambda_{n-k} = \max_{\mathbf{x} \neq 0, \mathbf{x} \perp \mathbf{v}_n, \ldots \mathbf{v}_{n-k+1}} \frac{\mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}}{\mathbf{x}^\mathsf{T}\mathbf{x}} = \max_{\mathbf{x}^\mathsf{T}\mathbf{x}=1, \mathbf{x} \perp \mathbf{v}_n, \ldots \mathbf{v}_{n-k+1}} \mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}$$

Spectral Clustering: Relaxing Balanced Cuts

objective function of spectral clustering

$$\min_{\mathbf{f}} \mathbf{f}^{\mathsf{T}}\mathbf{L}\mathbf{f} \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad \mathbf{f} \perp \mathbf{1}_n, \quad \|\mathbf{f}\| = \sqrt{n}$$

We have a solution: **second eigenvector**

How do we get the clustering?

The solution may not be integer. What to do?

$$\mathrm{cluster}_i = \begin{cases} 1 & \text{if } f_i \geq 0, \\ -1 & \text{if } f_i < 0. \end{cases}$$

Works but often too simple. In practice: cluster $\mathbf{f}$ using k-means to get $\{C_i\}_i$ and assign:

$$\mathrm{cluster}_i = \begin{cases} 1 & \text{if } i \in C_1, \\ -1 & \text{if } i \in C_{-1}. \end{cases}$$

Spectral Clustering: Approximating RatioCut

Wait, but we did not care about approximating mincut!

RatioCut

$$\mathrm{RatioCut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{|A|} + \frac{1}{|B|} \right)$$

Define graph function $\mathbf{f}$ for cluster membership of RatioCut:

$$f_i = \begin{cases} \sqrt{\frac{|B|}{|A|}} & \text{if } V_i \in A, \\ -\sqrt{\frac{|A|}{|B|}} & \text{if } V_i \in B. \end{cases}$$

$$\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i,j} w_{i,j}(f_i - f_j)^2 = (|A| + |B|)\mathrm{RatioCut}(A, B)$$

Spectral Clustering: Approximating RatioCut

Define graph function $\mathbf{f}$ for cluster membership of RatioCut:

$$f_i = \begin{cases} \sqrt{\frac{|B|}{|A|}} & \text{if } V_i \in A, \\ -\sqrt{\frac{|A|}{|B|}} & \text{if } V_i \in B. \end{cases}$$

$$\sum_i f_i = 0$$

$$\sum_i f_i^2 = n$$

objective function of spectral clustering (same)

$$\min_{\mathbf{f}} \mathbf{f}^\mathsf{T} \mathbf{L} \mathbf{f} \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad \mathbf{f} \perp \mathbf{1}_n, \quad \|\mathbf{f}\| = \sqrt{n}$$

Spectral Clustering: Approximating NCut

Normalized Cut

$$\mathrm{NCut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right)$$

Define graph function **f** for cluster membership of NCut:

$$f_i = \begin{cases} \sqrt{\frac{\mathrm{vol}(A)}{\mathrm{vol}(B)}} & \text{if } V_i \in A, \\ -\sqrt{\frac{\mathrm{vol}(B)}{\mathrm{vol}(A)}} & \text{if } V_i \in B. \end{cases}$$

$$(\mathbf{Df})^\mathsf{T} \mathbf{1}_n = 0 \qquad \mathbf{f}^\mathsf{T} \mathbf{Df} = \mathrm{vol}(V) \qquad \mathbf{f}^\mathsf{T} \mathbf{Lf} = \mathrm{vol}(V) \mathrm{NCut}(A, B)$$

objective function of spectral clustering (NCut)

$$\min_{\mathbf{f}} \mathbf{f}^\mathsf{T} \mathbf{Lf} \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad \mathbf{Df} \perp \mathbf{1}_n, \quad \mathbf{f}^\mathsf{T} \mathbf{Df} = \mathrm{vol}(V)$$

Spectral Clustering: Approximating NCut

objective function of spectral clustering (NCut)

$$\min_{\mathbf{f}} \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad \mathbf{D}\mathbf{f} \perp \mathbf{1}_n, \quad \mathbf{f}^\mathsf{T}\mathbf{D}\mathbf{f} = \mathrm{vol}(V)$$

Can we apply Rayleigh-Ritz now? Define $\mathbf{w} = \mathbf{D}^{1/2}\mathbf{f}$

objective function of spectral clustering (NCut)

$$\min_{\mathbf{w}} \mathbf{w}^\mathsf{T}\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{w} \quad \text{s.t.} \quad w_i \in \mathbb{R}, \mathbf{w} \perp \mathbf{D}^{1/2}\mathbf{1}_n, \|\mathbf{w}\|^2 = \mathrm{vol}(V)$$

objective function of spectral clustering (NCut)

$$\min_{\mathbf{w}} \mathbf{w}^\mathsf{T}\mathbf{L}_{\mathrm{sym}}\mathbf{w} \quad \text{s.t.} \quad w_i \in \mathbb{R}, \quad \mathbf{w} \perp \mathbf{v}_{1,\mathbf{L}_{\mathrm{sym}}}, \quad \|\mathbf{w}\|^2 = \mathrm{vol}(V)$$

Spectral Clustering: Approximating NCut

objective function of spectral clustering (NCut)

$$\min_{\mathbf{w}} \mathbf{w}^\mathsf{T} \mathbf{L}_{\mathrm{sym}} \mathbf{w} \quad \text{s.t.} \quad w_i \in \mathbb{R}, \quad \mathbf{w} \perp \mathbf{v}_{1,\mathbf{L}_{\mathrm{sym}}}, \quad \|\mathbf{w}\| = \mathrm{vol}(V)$$

Solution by Rayleigh-Ritz? $\mathbf{w} = \mathbf{v}_{2,\mathbf{L}_{\mathrm{sym}}}$ $\mathbf{f} = \mathbf{D}^{-1/2}\mathbf{w}$

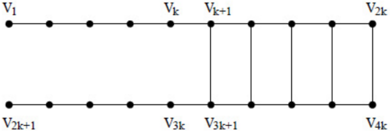
$\mathbf{f}$ is a the second eigenvector of $\mathbf{L}_{\mathrm{rw}}$!

tl;dr: Get the second eigenvector of $\mathbf{L}/\mathbf{L}_{\mathrm{rw}}$ for RatioCut/NCut.

Spectral Clustering: Approximation

These are all approximations. How bad can they be?

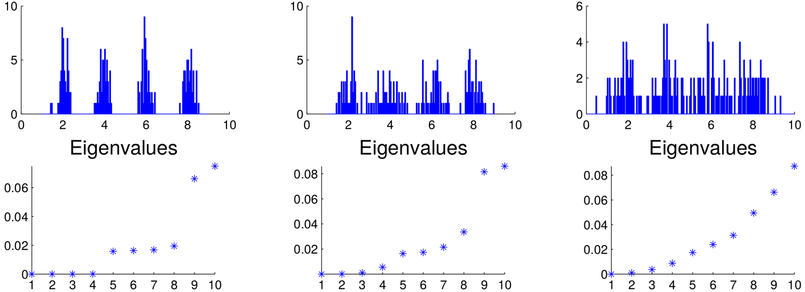
Example: cockroach graphs



No efficient approximation exist. Other relaxations possible.

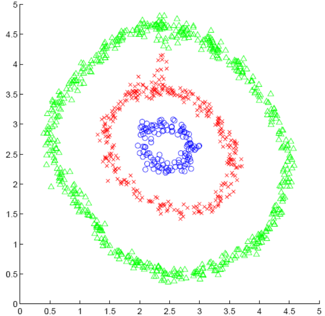
Spectral Clustering: 1D Example

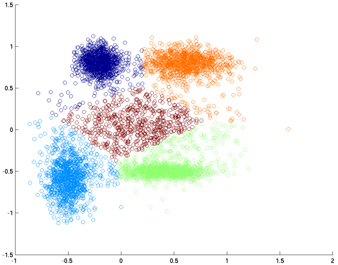
Elbow rule/EigenGap heuristic for number of clusters



Spectral Clustering: Understanding

Compactness vs. Connectivity

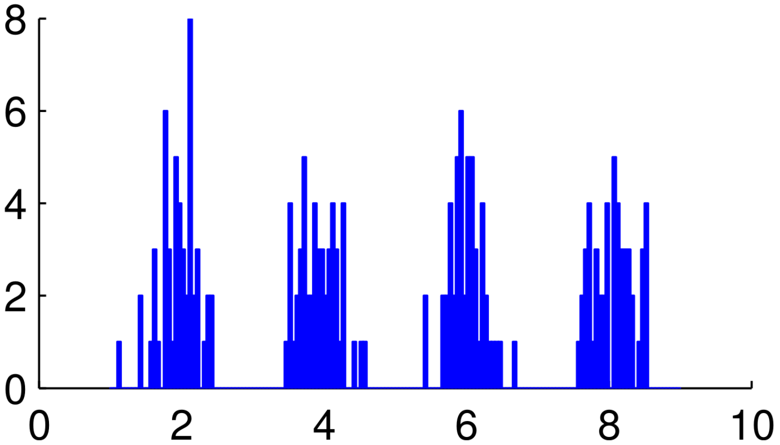




For which kind of date we can use one vs. the other?

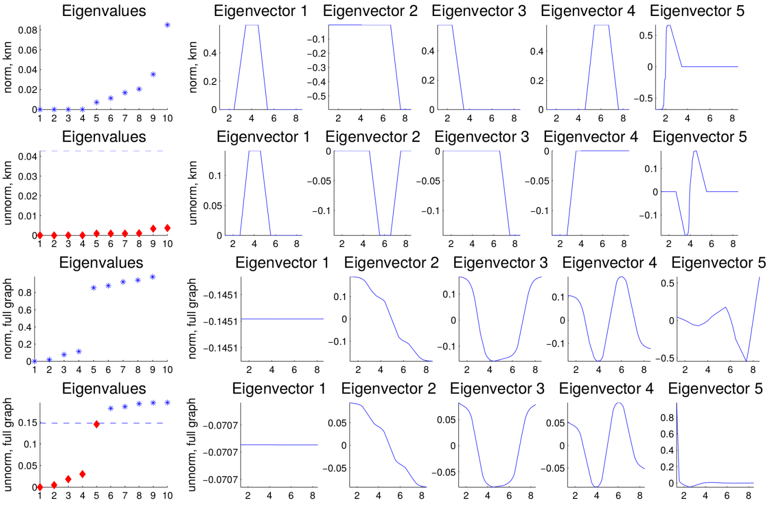
Any disadvantages of spectral clustering?

Spectral Clustering: 1D Example - Histogram



http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/publications/Luxburg07_tutorial.pdf

Spectral Clustering: 1D Example - Eigenvectors



Spectral Clustering: Bibliography

- M. Meila et al. "A random walks view of spectral segmentation". In: *AI and Statistics (AISTATS)* 57 (2001), p. 5287
- $\mathbf{L}_{\mathrm{sym}}$ Andrew Y Ng, Michael I Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in Neural Information Processing Systems 14*. 2001, pp. 849–856
- $\mathbf{L}_{\mathrm{rm}}$ J Shi and J Malik. "Normalized Cuts and Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), pp. 888–905
- Things can go wrong with the relaxation: Daniel A. Spielman and Shang H. Teng. "Spectral partitioning works: Planar graphs and finite element meshes". In: *Linear Algebra and Its Applications* 421 (2007), pp. 284–305

Michal Valko
michal.valko@inria.fr
sequel.lille.inria.fr