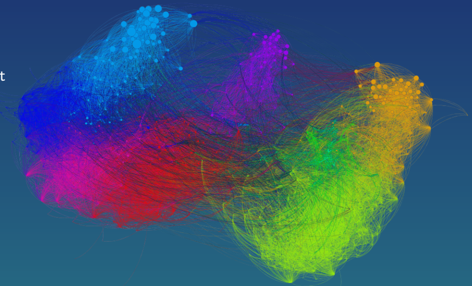
Graphs in Machine Learning

Michal Valko

DeepMind Paris and Inria Lille

TA: Omar Darwiche Domingues with the help of Pierre Perrault

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



Previous lecture

- where do the graphs come from?
  - social, information, utility, and biological networks
  - we create them from the flat data
  - random graph models
- specific applications and concepts
  - maximizing influence on a graph **gossip propagation, submodularity**, proof of the approximation guarantee
  - Google pagerank **random surfer process, steady state vector, sparsity**
  - online semi-supervised learning **label propagation, backbone graph, online learning, combinatorial sparsification, stability analysis**
  - Erdős number project, real-world graphs, **heavy tails, small world** – when did this happen?
- similarity graphs
  - different types
  - construction
  - practical considerations

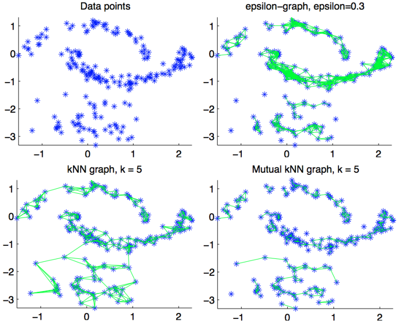
This Lecture

- spectral graph theory
- Laplacians and their properties
  - symmetric and asymmetric normalization
  - random walks
- geometry of the data and the connectivity
- spectral clustering
- manifold learning with Laplacians eigenmaps
- recommendation on a bipartite graph
- resistive networks
  - recommendation score as a resistance?
  - Laplacian and resistive networks
  - resistance distance and random walks
- PS: some students have started working on their projects already

Next Class: Lab Session

- 15. 10. 2019 by Omar (and Pierre)
- 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

Similarity Graphs: ε or k-NN?



http://www.ml.uni-saarland.de/code/GraphDemo/DemoSpectralClustering.htm

http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf

Generic Similarity Functions

Gaussian similarity function/Heat function/RBF:

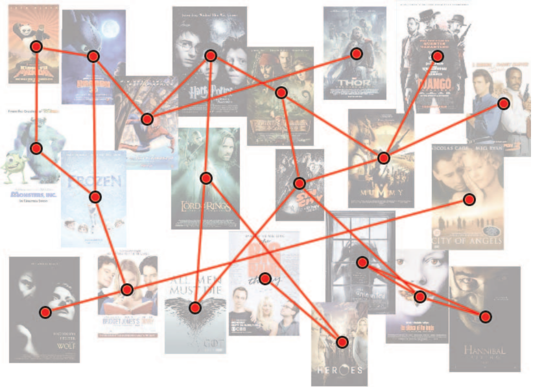
$$s_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

Cosine similarity function:

$$s_{ij} = \cos(\theta) = \left(\frac{\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}\right)$$

Typical Kernels

Similarity Graphs



$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ - with a set of **nodes** $\mathcal{V}$ and a set of **edges** $\mathcal{E}$

Sources of Real Networks

- **http://snap.stanford.edu/data/**
- http://www-personal.umich.edu/~mejn/netdata/
- http://proj.ise.bgu.ac.il/sns/datasets.html
- http://www.cise.ufl.edu/research/sparse/matrices/
- http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

graph Laplacian

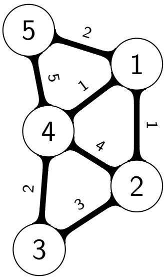
...the only matrix that matters

Graph Laplacian

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ - with a set of **nodes** $\mathcal{V}$ and a set of **edges** $\mathcal{E}$

$\mathbf{A}$	adjacency matrix
$\mathbf{W}$	weight matrix
$\mathbf{D}$	(diagonal) degree matrix
$\mathbf{L} = \mathbf{D} - \mathbf{W}$	graph **Laplacian** matrix

$$\mathbf{L} = \begin{pmatrix} 4 & -1 & 0 & -1 & -2 \\ -1 & 8 & -3 & -4 & 0 \\ 0 & -3 & 5 & -2 & 0 \\ -1 & -4 & -2 & 12 & -5 \\ -2 & 0 & 0 & -5 & 7 \end{pmatrix}$$



L is SDD!

demo: `https://dominikschmidt.xyz/spectral-clustering-exp/`

Properties of Graph Laplacian

Graph function: a vector $\mathbf{f} \in \mathbb{R}^N$ assigning values to nodes:

$$\mathbf{f} : \mathcal{V}(\mathcal{G}) \to \mathbb{R}.$$

$$\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i,j \leq N} w_{i,j}(f_i - f_j)^2 = S_G(\mathbf{f})$$

Recap: Eigenwerte und Eigenvektoren

A vector $\mathbf{v}$ is an **eigenvector** of matrix $\mathbf{M}$ of **eigenvalue** λ

$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v}.$$

If $(\lambda_1, \mathbf{v}_1)$ are $(\lambda_2, \mathbf{v}_2)$ **eigenpairs** for symmetric $\mathbf{M}$ with $\lambda_1 \neq \lambda_2$ then $\mathbf{v}_1 \perp \mathbf{v}_2$, i.e., $\mathbf{v}_1^\mathsf{T}\mathbf{v}_2 = 0$.

If $(\lambda, \mathbf{v}_1)$, $(\lambda, \mathbf{v}_2)$ are eigenpairs for $\mathbf{M}$ then $(\lambda, \mathbf{v}_1 + \mathbf{v}_2)$ is as well.

For symmetric $\mathbf{M}$, the **multiplicity** of λ is the dimension of the space of eigenvectors corresponding to λ.

$N \times N$ symmetric matrix has N eigenvalues (w/ multiplicities).

Eigenvalues, Eigenvectors, and Eigendecomposition

A vector $\mathbf{v}$ is an **eigenvector** of matrix $\mathbf{M}$ of **eigenvalue** λ

$$\mathbf{M}\mathbf{v} = \lambda \mathbf{v}.$$

Vectors $\{\mathbf{v}_i\}_i$ form an **orthonormal** basis with $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_N$.

$$\forall i \quad \mathbf{M}\mathbf{v}_i = \lambda_i \mathbf{v}_i \qquad \equiv \qquad \mathbf{M}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$$

$\mathbf{Q}$ has eigenvectors in columns and $\mathbf{\Lambda}$ has eigenvalues on its diagonal.

Right-multiplying $\mathbf{M}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$ by $\mathbf{Q}^\mathsf{T}$ we get the **eigendecomposition** of $\mathbf{M}$:

$$\mathbf{M} = \mathbf{M}\mathbf{Q}\mathbf{Q}^\mathsf{T} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathsf{T} = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^\mathsf{T}$$

$\mathbf{M} = \mathbf{L}$: Properties of Graph Laplacian

We can assume **non-negative weights**: $w_{ij} \geq 0$.

L is symmetric

L positive semi-definite $\leftarrow \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i,j\leq N} w_{i,j}(f_i - f_j)^2$

Recall: If $\mathbf{L}\mathbf{f} = \lambda\mathbf{f}$ then λ is an **eigenvalue** (of the Laplacian).

The smallest eigenvalue of **L** is 0. Corresponding eigenvector: $\mathbf{1}_N$.

All eigenvalues are non-negative reals $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$.

Self-edges do not change the value of **L**.

Properties of Graph Laplacian

The multiplicity of eigenvalue 0 of $\mathbf{L}$ equals to the number of connected components. The eigenspace of 0 is spanned by the components' indicators.

Proof: If $(0, \mathbf{f})$ is an eigenpair then $0 = \frac{1}{2}\sum_{i,j\leq N} w_{i,j}(f_i - f_j)^2$. Therefore, $\mathbf{f}$ is constant on each connected component. If there are k components, then $\mathbf{L}$ is k-block-diagonal:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & & & \\ & \mathbf{L}_2 & & \\ & & \ddots & \\ & & & \mathbf{L}_k \end{bmatrix}$$

For block-diagonal matrices: the spectrum is the union of the spectra of $\mathbf{L}_i$ (eigenvectors of $\mathbf{L}_i$ padded with zeros elsewhere).

For $\mathbf{L}_i$ $(0, \mathbf{1}_{|V_i|})$ is an eigenpair, hence the claim.

Smoothness of the Function and Laplacian

- $\mathbf{f} = (f_1, \ldots, f_N)^\mathsf{T}$: graph function
- Let $\mathbf{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathbf{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathbf{L}$.
  - Columns of $\mathbf{Q}$ form a basis.
- $\boldsymbol{\alpha}$: Unique vector such that $\mathbf{Q}\boldsymbol{\alpha} = \mathbf{f}$ Note: $\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}$

Smoothness of a graph function $S_G(\mathbf{f})$

$$S_G(\mathbf{f}) = \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \mathbf{f}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^{N} \lambda_i \alpha_i^2$$

Smoothness and regularization: Small value of

(a) $S_G(\mathbf{f})$ **(b)** $\boldsymbol{\Lambda}$ norm of $\boldsymbol{\alpha}^\star$ **(c)** $\alpha_i^\star$ for large λ_i

Smoothness of the Function and Laplacian

$$S_G(\mathbf{f}) = \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \mathbf{f}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^{N}\lambda_i\alpha_i^2$$

Eigenvectors are graph functions too!

What is the smoothness of an eigenvector?

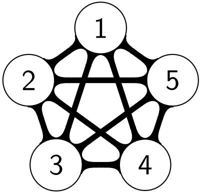
Spectral coordinates of eigenvector $\mathbf{v}_k$: $\mathbf{Q}^\mathsf{T}\mathbf{v}_k = \mathbf{e}_k$

$$S_G(\mathbf{v}_k) = \mathbf{v}_k^\mathsf{T}\mathbf{L}\mathbf{v}_k = \mathbf{v}_k^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\mathbf{v}_k = \mathbf{e}_k^\mathsf{T}\boldsymbol{\Lambda}\mathbf{e}_k = \|\mathbf{e}_k\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^{N}\lambda_i(\mathbf{e}_k)_i^2 = \lambda_k$$

The smoothness of k-th eigenvector is the k-th eigenvalue.

Laplacian of the Complete Graph K_N

What is the eigenspectrum of $\mathbf{L}_{K_N}$?



$$\mathbf{L}_{K_N} = \begin{pmatrix} N-1 & -1 & -1 & -1 & -1 \\ -1 & N-1 & -1 & -1 & -1 \\ -1 & -1 & N-1 & -1 & -1 \\ -1 & -1 & -1 & N-1 & -1 \\ -1 & -1 & -1 & -1 & N-1 \end{pmatrix}$$

From before: we know that $(0, \mathbf{1}_N)$ is an eigenpair.

If $\mathbf{v} \neq 0_N$ and $\mathbf{v} \perp \mathbf{1}_N \implies \sum_i \mathbf{v}_i = 0$. To get the other eigenvalues, we compute $(\mathbf{L}_{K_N}\mathbf{v})_1$ and divide by $\mathbf{v}_1$ (wlog $\mathbf{v}_1 \neq 0$).

$$(\mathbf{L}_{K_N}\mathbf{v})_1 = (N-1)\mathbf{v}_1 - \sum_{i=2}^{N} \mathbf{v}_i = N\mathbf{v}_1.$$

What are the remaining eigenvalues/vectors?

Normalized Laplacians

$$\begin{aligned}
\mathbf{L}_{un} &= \mathbf{D} - \mathbf{W} \\
\mathbf{L}_{sym} &= \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \\
\mathbf{L}_{rw} &= \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}
\end{aligned}$$

$$\mathbf{f}^{\mathsf{T}}\mathbf{L}_{sym}\mathbf{f} = \frac{1}{2}\sum_{i,j \leq N} w_{i,j}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2$$

$(\lambda, \mathbf{u})$ is an eigenpair for $\mathbf{L}_{rw}$ iff $(\lambda, \mathbf{D}^{1/2}\mathbf{u})$ is an eigenpair for $\mathbf{L}_{sym}$

Normalized Laplacians

$\mathbf{L}_{sym}$ and $\mathbf{L}_{rw}$ are PSD with non-negative real eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N$$

$(\lambda, \mathbf{u})$ is an eigenpair for $\mathbf{L}_{rw}$ iff $(\lambda, \mathbf{u})$ solve the generalized eigenproblem $\mathbf{L}\mathbf{u} = \lambda \mathbf{D}\mathbf{u}$.

$(0, \mathbf{1}_N)$ is an eigenpair for $\mathbf{L}_{rw}$.

$(0, \mathbf{D}^{1/2}\mathbf{1}_N)$ is an eigenpair for $\mathbf{L}_{sym}$.

Multiplicity of eigenvalue 0 of $\mathbf{L}_{rw}$ or $\mathbf{L}_{sym}$ equals to the number of connected components.

Proof: As for $\mathbf{L}$.

Laplacian and Random Walks on Undirected Graphs

- stochastic process: vertex-to-vertex jumping
- transition probability $v_i \to v_j$ is $p_{ij} = w_{ij}/d_i$
  - $d_i \stackrel{\text{def}}{=} \sum_j w_{ij}$
- transition matrix $\mathbf{P} = (p_{ij})_{ij} = \mathbf{D}^{-1}\mathbf{W}$ (notice $\mathbf{L}_{rw} = \mathbf{I} - \mathbf{P}$)
- if G is connected and non-bipartite $\to$ unique **stationary distribution** $\pi = (\pi_1, \pi_2, \pi_3, \ldots, \pi_N)$ where $\pi_i = d_i/\mathrm{vol}(V)$
  - $\mathrm{vol}(G) = \mathrm{vol}(V) = \mathrm{vol}(\mathbf{W}) \stackrel{\text{def}}{=} \sum_i d_i = \sum_{i,j} w_{ij}$
- $\pi = \frac{\mathbf{1}^\mathsf{T}\mathbf{W}}{\mathrm{vol}(\mathbf{W})}$ verifies $\pi\mathbf{P} = \pi$ as:

$$\pi\mathbf{P} = \frac{\mathbf{1}^\mathsf{T}\mathbf{W}\mathbf{P}}{\mathrm{vol}(\mathbf{W})} = \frac{\mathbf{1}^\mathsf{T}\mathbf{D}\mathbf{P}}{\mathrm{vol}(\mathbf{W})} = \frac{\mathbf{1}^\mathsf{T}\mathbf{D}\mathbf{D}^{-1}\mathbf{W}}{\mathrm{vol}(\mathbf{W})} = \frac{\mathbf{1}^\mathsf{T}\mathbf{W}}{\mathrm{vol}(\mathbf{W})} = \pi$$

What's the difference from the PageRank™?

$$\mathrm{cut}(A, B) = \frac{1}{2}\mathbf{f}^{\top}\mathbf{L}\mathbf{f}$$

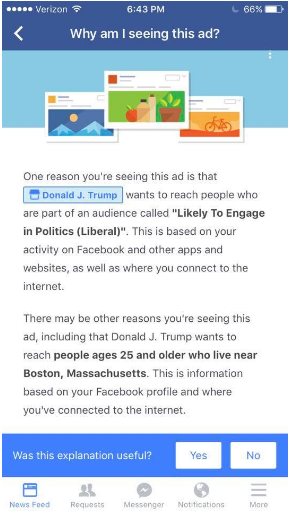
spectral clustering

...with connectivity beyond compactness

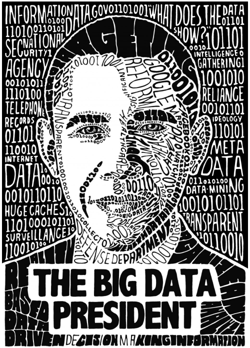
How to rule the world?

Let's make France great again!

How to rule the world?



How to rule the world: “AI” is here

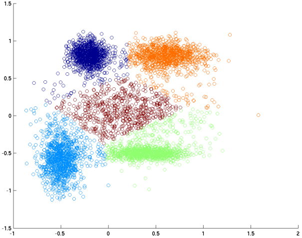


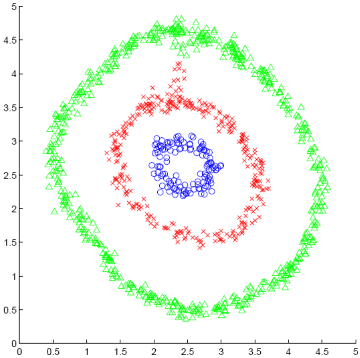
https://www.washingtonpost.com/opinions/obama-the-big-data-president/2013/06/14/1d71fe2e-d391-11e2-b05f-3ea3f0e7bb5a_story.html

https://www.technologyreview.com/s/509026/how-obamas-team-used-big-data-to-rally-voters/

Talk of Rayid Ghaniy: https://www.youtube.com/watch?v=gDM1GuszM_U

Application of Graphs for ML: Clustering

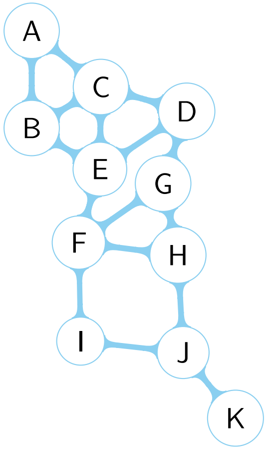




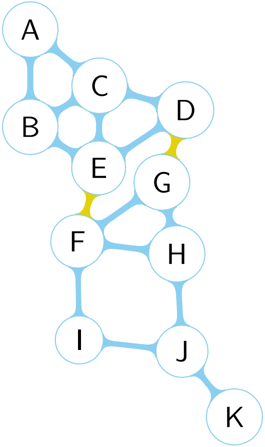
Application: Clustering - Recap

- What do we know about the **clustering** in general?
  - ill defined problem (different tasks $\rightarrow$ different paradigms)
  - "I know it when I see it"
  - 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

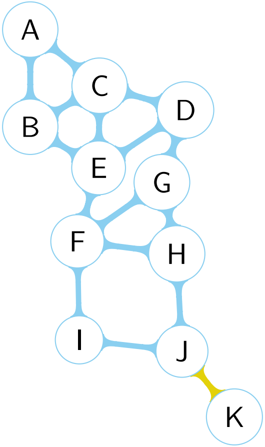


Spectral Clustering: Cuts on graphs



Defining the cut objective we get the clustering!

Spectral Clustering: Cuts on graphs



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

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

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

Easily generalizable to $k \geq 2$

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

Approximate!

Spectral Clustering: Relaxing Balanced Cuts

Relaxation for (simple) balanced cuts for 2 sets

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

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 $\mathbf{L}$ then

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

$$\lambda_N = \max_{\mathbf{x}\neq 0} \frac{\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}}{\mathbf{x}^\mathsf{T}\mathbf{x}} = \max_{\mathbf{x}^\mathsf{T}\mathbf{x}=1} \mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}$$

$\frac{\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}}{\mathbf{x}^\mathsf{T}\mathbf{x}} \equiv$ **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 (Courant-Fischer-Weyl)

If $\lambda_1 \leq \cdots \leq \lambda_N$ are the eigenvectors of real symmetric $\mathbf{L}$ and $\mathbf{v}_1, \ldots, \mathbf{v}_N$ the corresponding orthogonal eigenvalues, then for $k = 1 : N - 1$

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

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

Rayleigh-Ritz theorem: Quick and dirty proof

When we reach the extreme points?

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}}{\mathbf{x}^\mathsf{T}\mathbf{x}}\right) = \frac{\partial}{\partial \mathbf{x}}\left(\frac{f(\mathbf{x})}{g(\mathbf{x})}\right) = 0 \iff f'(\mathbf{x})g(\mathbf{x}) = f(\mathbf{x})g'(\mathbf{x})$$

By matrix calculus (or just calculus):

$$\frac{\partial \mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{L}\mathbf{x} \quad \text{and} \quad \frac{\partial \mathbf{x}^\mathsf{T}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{x}$$

When $f'(\mathbf{x})g(\mathbf{x}) = f(\mathbf{x})g'(\mathbf{x})$?

$$\mathbf{L}\mathbf{x}\left(\mathbf{x}^\mathsf{T}\mathbf{x}\right) = \left(\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}\right)\mathbf{x} \iff \mathbf{L}\mathbf{x} = \frac{\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}}{\mathbf{x}^\mathsf{T}\mathbf{x}}\mathbf{x} \iff \mathbf{L}\mathbf{x} = \lambda\mathbf{x}$$

Conclusion: Extremes are the eigenvectors with their eigenvalues

Spectral Clustering: Relaxing Balanced Cuts

objective function of spectral clustering

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

Solution: **second eigenvector** How do we get the clustering?

The solution may not be integral. What to do?

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

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

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

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 **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 - it's magic!)

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

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 $\mathbf{f}$ for cluster membership of NCut:

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

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

objective function of spectral clustering (NCut)

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

Spectral Clustering: Approximating NCut

objective function of spectral clustering (NCut)

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

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

objective function of spectral clustering (NCut)

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

objective function of spectral clustering (NCut)

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

Spectral Clustering: Approximating NCut

objective function of spectral clustering (NCut)

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

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 $\mathbf{L}_{\mathrm{rw}}$!

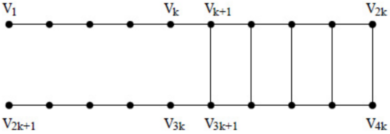
tl;dr: Get the second eigenvector of $\mathbf{L}/\mathbf{L}_{\mathrm{rw}}$ for RatioCut/NCut.

demo: `https://dominikschmidt.xyz/spectral-clustering-exp/`

Spectral Clustering: Approximation

These are all approximations. How bad can they be?

Example: cockroach graphs

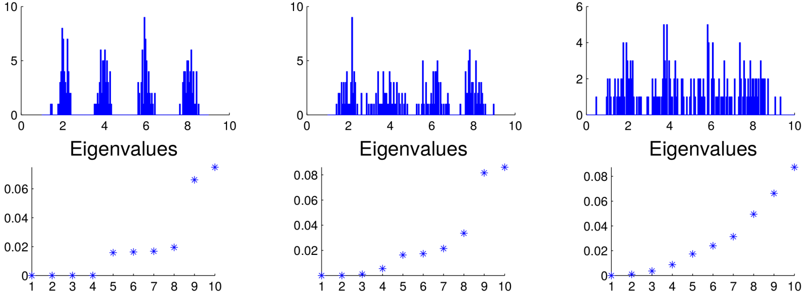


No efficient approximation exist. Other relaxations possible.

https://www.cs.cmu.edu/~glmiller/Publications/Papers/GuMi95.pdf

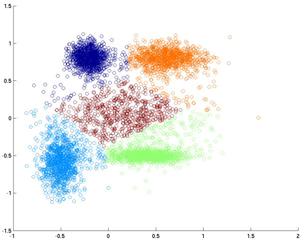
Spectral Clustering: 1D Example

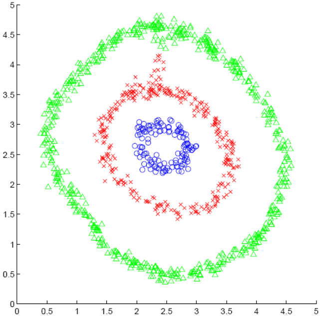
Elbow rule/EigenGap heuristic for number of clusters



Spectral Clustering: Understanding

Compactness vs. **Connectivity**

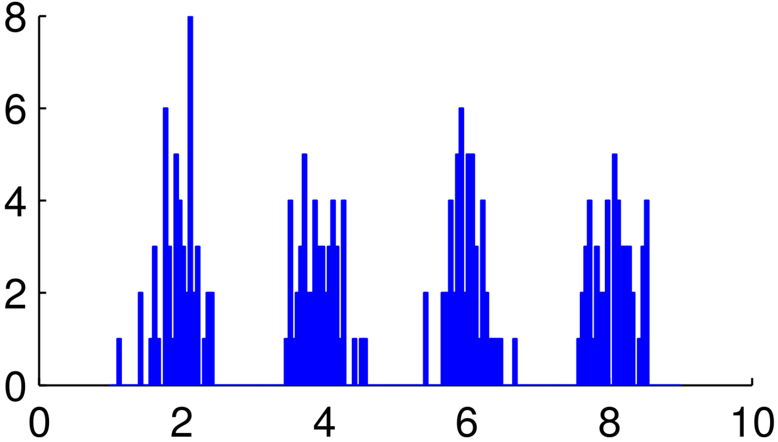




For which kind of data we can use one vs. the other?

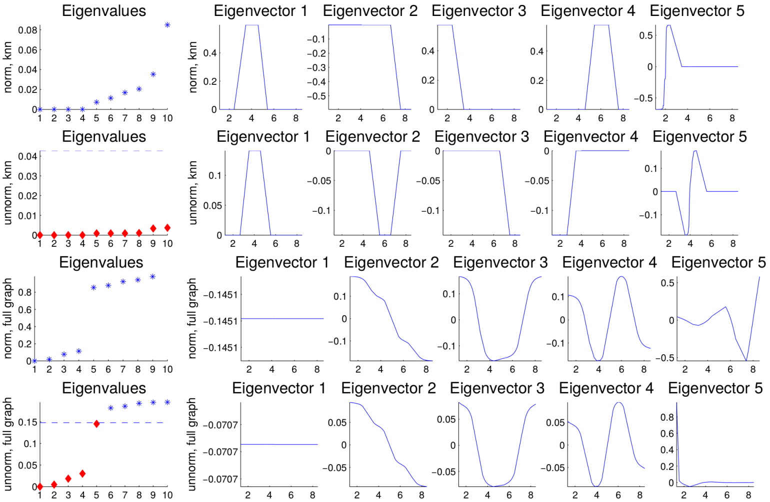
Any disadvantages of spectral clustering?

Spectral Clustering: 1D Example - Histogram



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

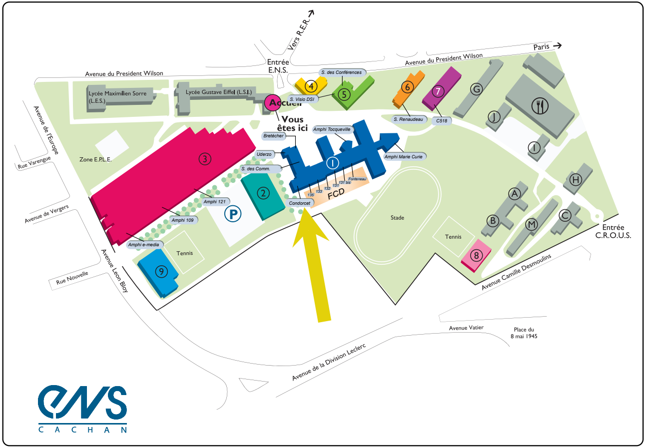
Spectral Clustering: 1D Example - Eigenvectors



Spectral Clustering: Bibliography

- M. Meila et al. "A random walks view of spectral segmentation". In: *International Conference on Artificial Intelligence and Statistics* (2001)
- $\mathbf{L}_{\mathrm{sym}}$ Andrew Y Ng, Michael I Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Neural Information Processing Systems*. 2001
- $\mathbf{L}_{\mathrm{rm}}$ 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

Next class on Tuesday, October 22th at 13:30!



Michal Valko
contact via Piazza