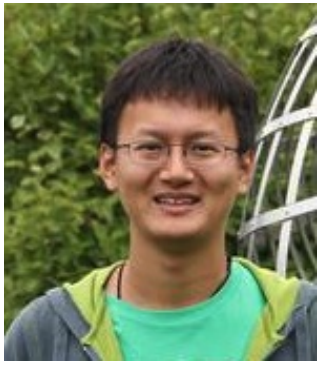
A rigorous framework for data clustering

Thomas Strohmer
Department of Mathematics
University of California, Davis

Statistics Seminar Series
UC Davis, Nov. 7, 2019

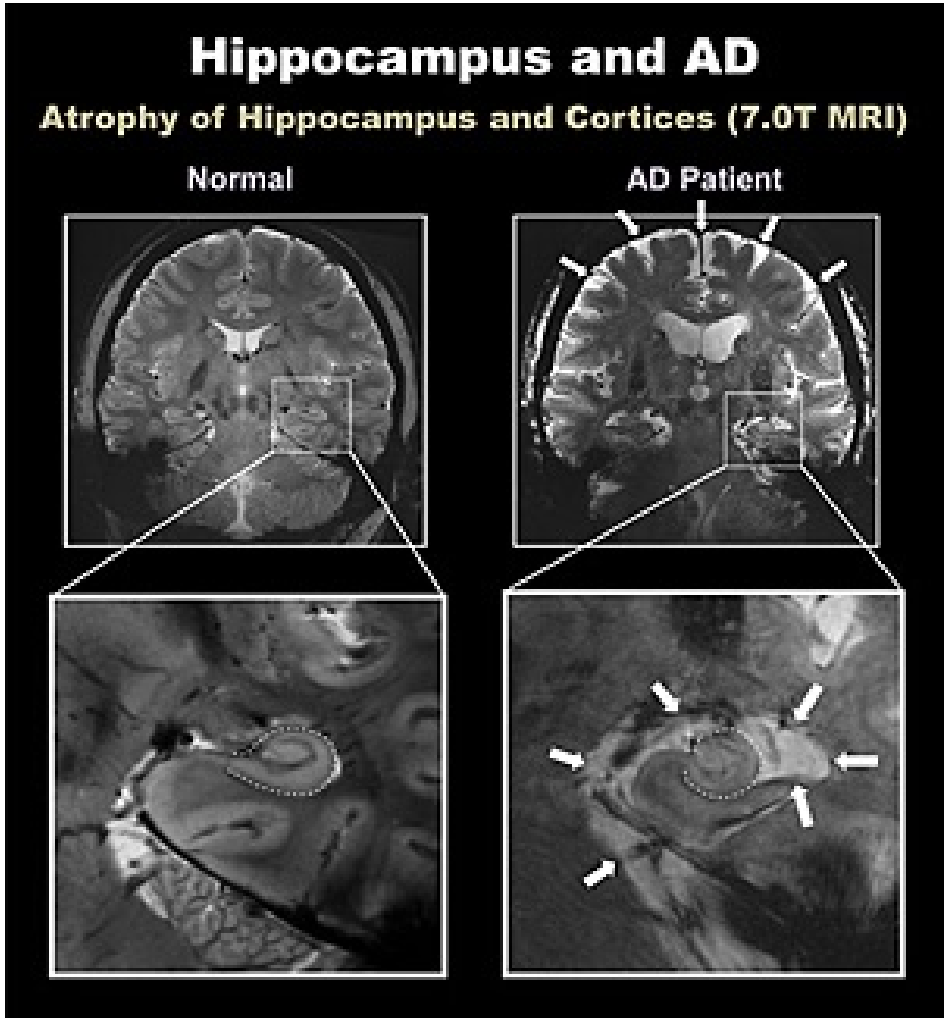
Acknowledgements

Joint work with Shuyang Ling (NYU)

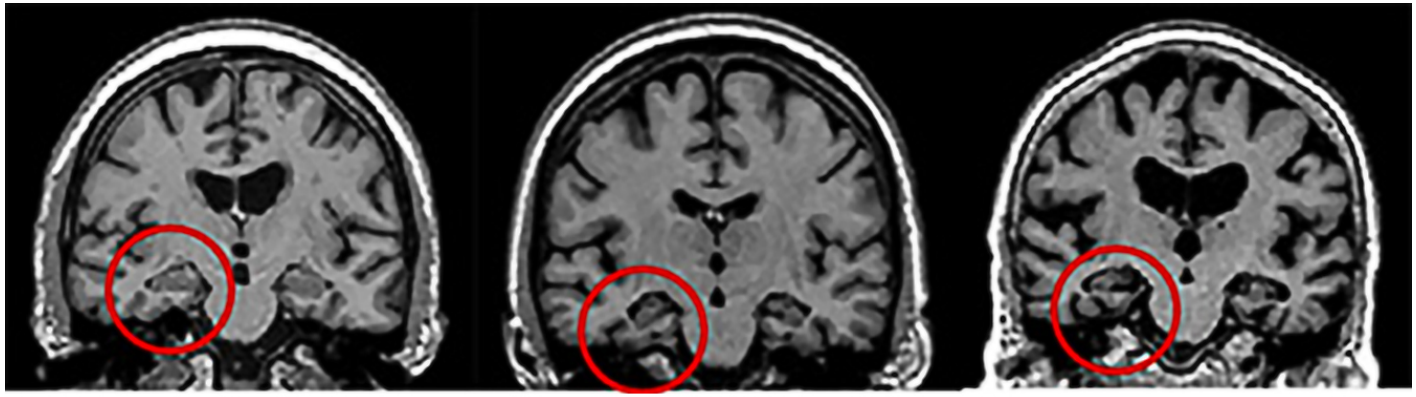


This work is sponsored by the NSF-DMS and the National Geospatial-Intelligence Agency.

Trying to detect Alzheimer's from MRIs

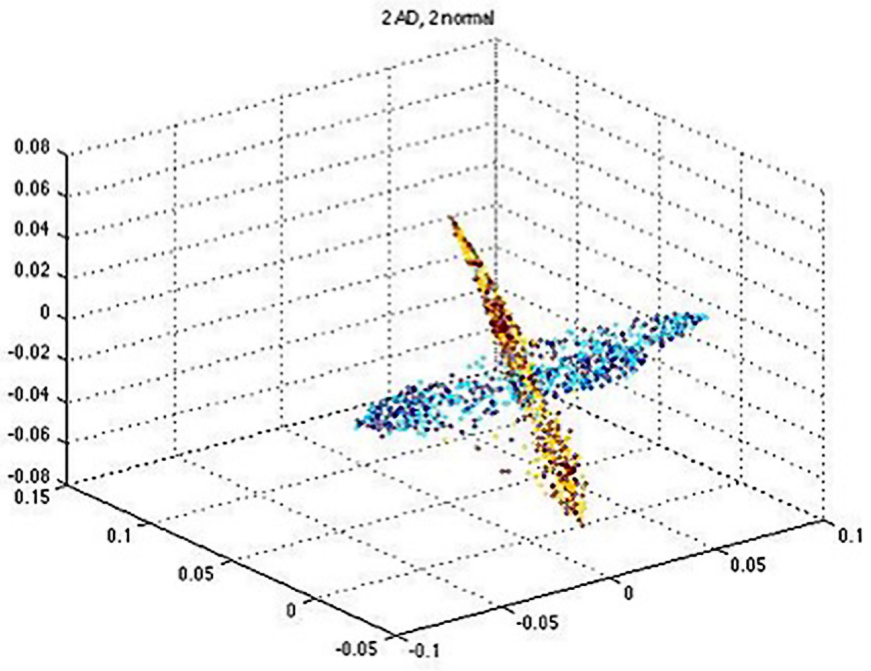


Trying to detect Alzheimer's from MRIs



75 year old Control | 75 year old MCI | 75 year old AD

Unsupervised Learning via diffusion maps



Diffusion component representation of two normal brains (dark blue and light blue) and two Alzheimer brains (red, yellow)

Unsupervised Learning via diffusion maps

Sometimes our diffusion component analysis approach worked very well, and sometimes it failed. Why?

Alzheimer MRI data are too complicated to develop a thorough theoretical analysis.

Look at existing theory for simpler situations.

Unsupervised Learning via diffusion maps

Sometimes our diffusion component analysis approach worked very well, and sometimes it failed. Why?

Alzheimer MRI data are too complicated to develop a thorough theoretical analysis.

Look at existing theory for simpler situations.

Surprise:
No useful theory about performance of diffusion maps.

Unsupervised Learning via diffusion maps

Sometimes our diffusion component analysis approach worked very well, and sometimes it failed. Why?

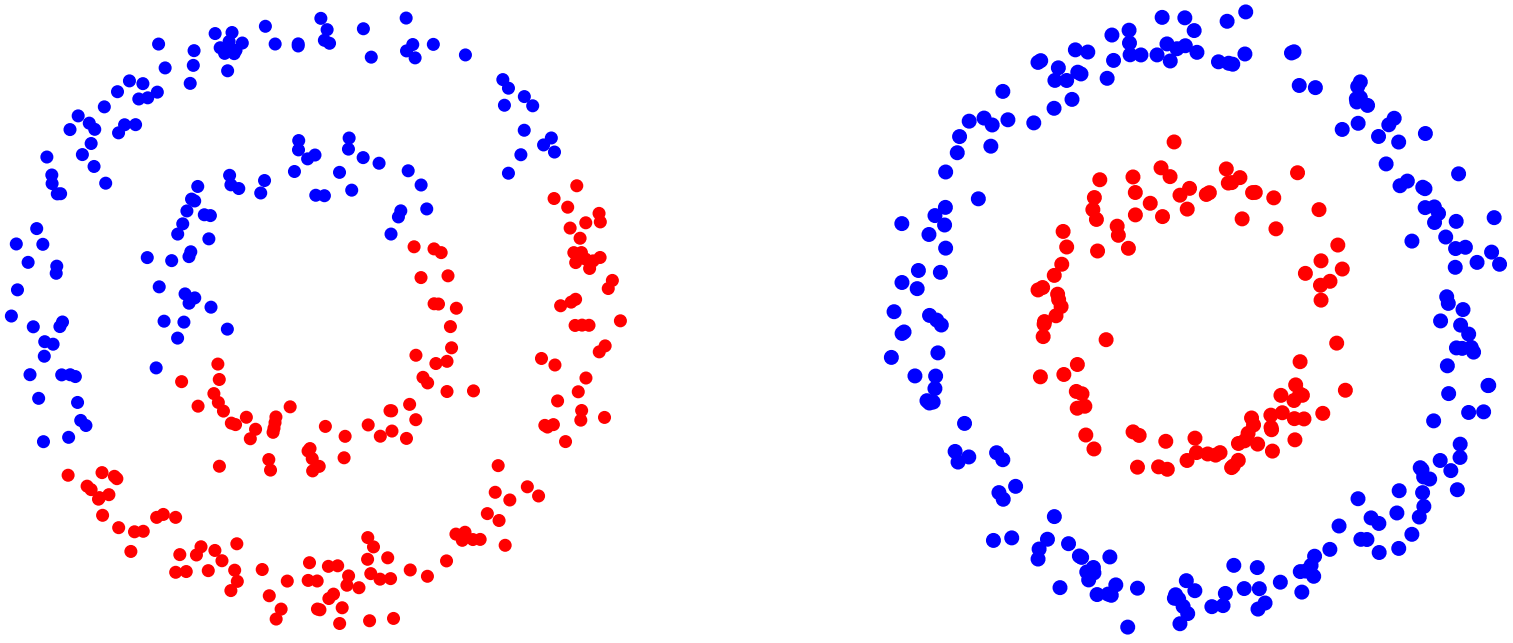
Alzheimer MRI data are too complicated to develop a thorough theoretical analysis.

Look at existing theory for simpler situations.

Surprise:
No useful theory about performance of diffusion maps.

Bigger surprise:
There is no useful theory about performance of spectral clustering, despite the fact that it has been around for decades!

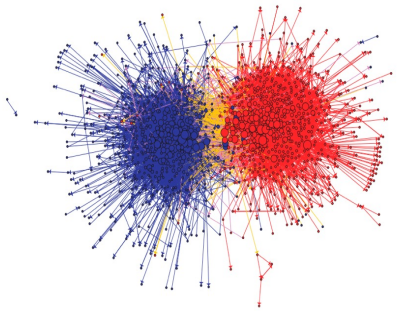
Spectral clustering: learning the “shape” of data



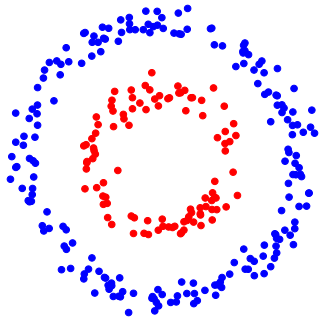
Existing theory for the performance of spectral clustering either assumes that the clusters are infinitely far apart or that the associated graph has disconnected components as expressed in a “perfect” binary weight matrix.

Spectral clustering, graph cuts, community detection

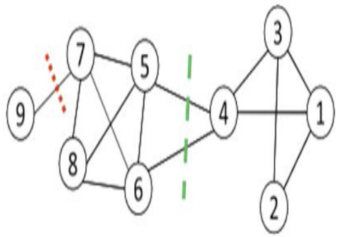
Community detection



Spectral clustering



Graph cuts



Three NP-hard problems ...

Data clustering and unsupervised learning

Question: Given a set of N data points in $\mathbb{R}^d$, how to partition them into k clusters based on their similarity?

K-means: minimize the total within-cluster sum of squared error to estimate the partition.

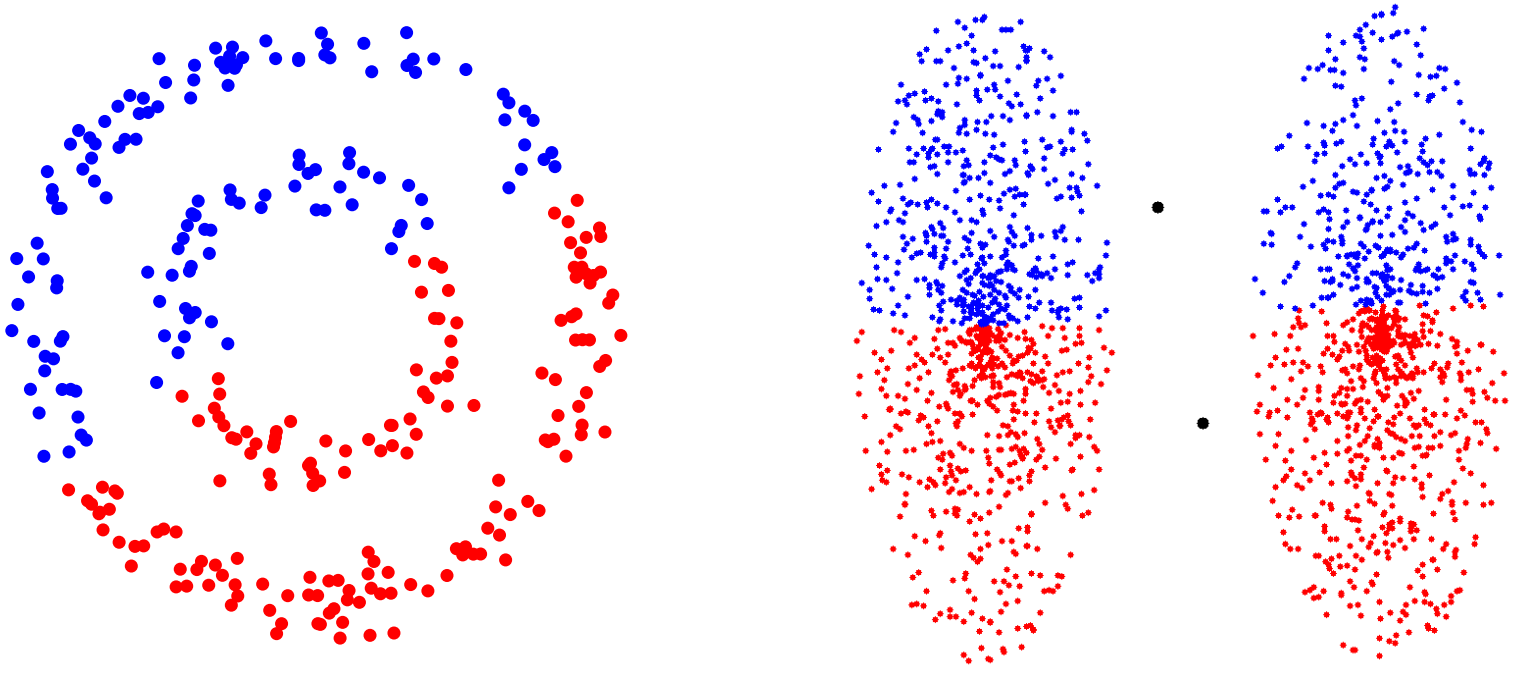
$$\min_{\{\Gamma_l\}_{l=1}^k} \underbrace{\sum_{l=1}^{k} \sum_{i\in\Gamma_l} \left\| x_i - \overbrace{\frac{1}{|\Gamma_l|} \sum_{i\in\Gamma_l} x_i}^{\text{centroid}} \right\|^2}_{\text{within-cluster sum of squares}}$$

where $\{\Gamma_l\}_{l=1}^k$ is a partition of $\{1, \cdots, N\}$.

Limitation of *k*-means

K-means only works for datasets with individual clusters:

isotropic and within convex boundaries, well-separated



Kernel k-means and nonlinear embedding

Goal: map the data into a feature space so that they are well-separated and k-means would work.

How: locally-linear embedding, isomap, multidimensional scaling, Laplacian eigenmaps, diffusion maps, etc.

Focus: We will focus on Laplacian eigenmaps. Spectral clustering consists of Laplacian eigenmaps followed by k-means clustering.

Spectral clustering

Spectral clustering[1] consists of two steps:

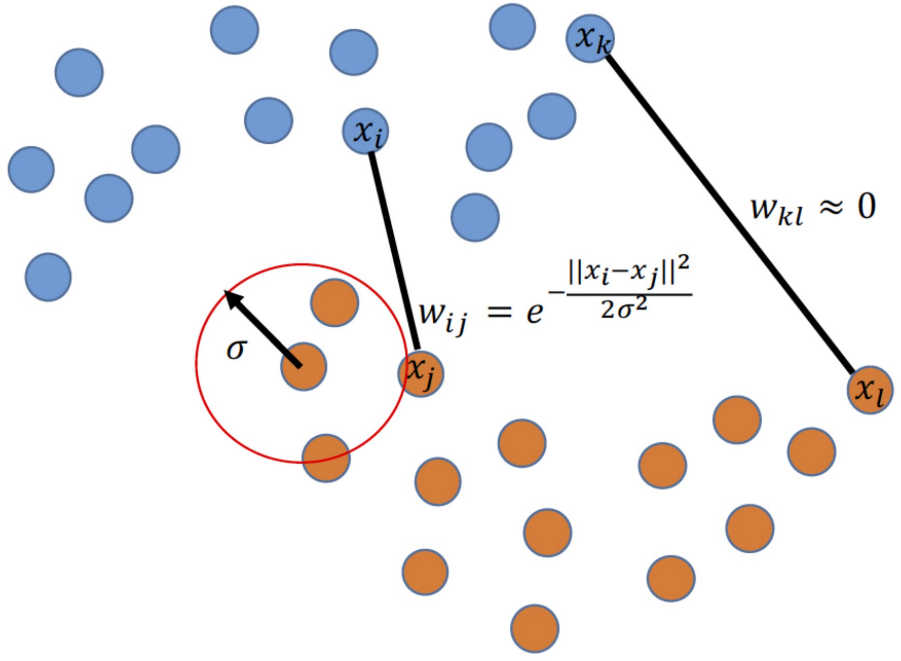
- Laplacian eigenmaps
- “Rounding” procedure (e.g. k-means)

[1] [Luxburg, 07], [Shi, Malik, 02], [Belkin, Niyogi, 03]

The Graph Laplacian

One key ingredient of spectral clustering is the graph Laplacian.

Given $\{\boldsymbol{x}_i\}_{i=1}^N \in \mathbb{R}^d$, we construct a similarity (weight) matrix $\boldsymbol{W}$.



The graph Laplacian is

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

where $\boldsymbol{D} = \text{diag}(\boldsymbol{W}\mathbf{1}_N)$ is the degree matrix.

The spectra of L are related to the graph's connectivity, especially $\lambda_2(\boldsymbol{L})$.

The Graph Laplacian

The *incidence matrix* ∇ of a graph is

$$\nabla := \begin{cases} \nabla = -1 & \text{if } v \text{ is the initial vertex of the edge} \\ \nabla = 1 & \text{if } v \text{ is the terminal vertex of the edge} \\ \nabla = 0 & \text{if } v \text{ is not in the edge,} \end{cases}$$

where we assume each edge has an arbitrary (but fixed) orientation. If we have an undirected graph, we can obtain the incidence matrix by choosing a (fixed) orientation of the edges. If f is a function acting on the vertices, then

$$\nabla f = \{f(v_j) - f(v_i)\}_{i,j}$$

Hence ∇ is a kind of difference operator.
The unweighted graph Laplacian $\boldsymbol{L}$ can be written as $\boldsymbol{L} = \nabla^T \nabla$.

Laplacian eigenmap

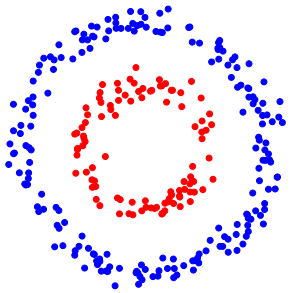
Denote the eigenvectors of $\boldsymbol{L}$ by $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N$ with associated real, non-negative eigenvalus $\lambda_1, \cdots, \lambda_N$.

The Laplacian eigenmap is defined as

$$\begin{bmatrix} \varphi(\boldsymbol{x}_1) \\ \vdots \\ \varphi(\boldsymbol{x}_N) \end{bmatrix} := \underbrace{[\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n]}_{\boldsymbol{U}} \in \mathbb{R}^{N \times n}$$

where $\{\boldsymbol{u}_l\}_{l=1}^{n}$ are the eigenvectors w.r.t. the n **smallest** eigenvalues.

Laplacian eigenmaps, k-means, spectral clustering



$\xrightarrow{\varphi\text{: nonlinear map}}$

φ maps data in $\mathbb{R}^d$ to $\mathbb{R}^n$; coordinates in terms of eigenvectors:

$$\varphi : \underbrace{\boldsymbol{x}_i}_{\mathbb{R}^d} \longrightarrow \underbrace{\varphi(\boldsymbol{x}_i)}_{\mathbb{R}^n}.$$

Then we apply k-means to $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^N$ to perform clustering.

Comments on spectral clustering

Pros and Cons of spectral clustering

Pros:

- Spectral clustering enjoys high popularity and conveniently applies to various settings.
- Supposedly works better than vanilla k-means

Cons:

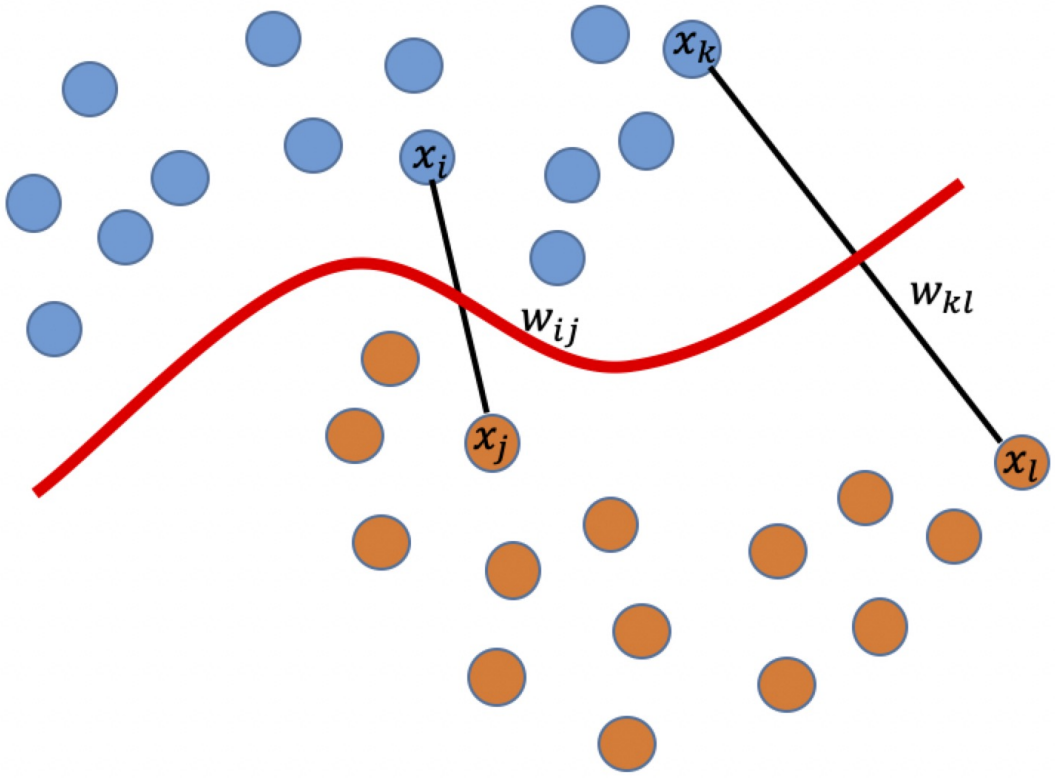
- Rigorous justification of spectral clustering is still lacking.
- Two-step procedure complicates the theoretic analysis.

Our goal:

- A different route: **convex relaxation** of spectral clustering.
- Establish a theoretical framework for spectral clustering.

A graph cut perspective

- The matrix ***W*** is viewed as a weight matrix of a graph with *N* vertices.



- Partitioning the dataset into *k* clusters is equivalent to finding a *k*-way graph cut such that any pair of induced subgraphs is not well-connected.

A graph cut perspective

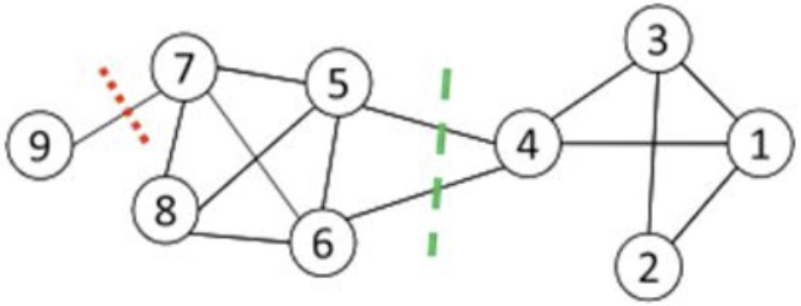
Graph cut

The cut is defined as the weight sum of edges whose two ends are in different subsets,

$$\mathrm{cut}(\Gamma, \Gamma^c) := \sum_{i \in \Gamma, j \in \Gamma^c} w_{ij}$$

where Γ is a subset of vertices and Γ^c is its complement.

Warning: unbalanced cuts!



RatioCut

The ratio cut induced by the partition $\{\Gamma_a\}_{a=1}^k$ is equal to

$$\mathrm{RatioCut}(\{\Gamma_a\}_{a=1}^k) = \sum_{a=1}^{k} \frac{\mathrm{cut}(\Gamma_a, \Gamma_a^c)}{|\Gamma_a|}.$$

In particular, if $k = 2$,

$$\mathrm{RatioCut}(\Gamma, \Gamma^c) = \frac{\mathrm{cut}(\Gamma, \Gamma^c)}{|\Gamma|} + \frac{\mathrm{cut}(\Gamma, \Gamma^c)}{|\Gamma^c|}.$$

Minimizing RatioCut is NP-hard in general!

RatioCut and the graph Laplacian

Let $\mathbf{1}_{\Gamma_a}(\cdot)$ be an indicator vector which maps a vertex to a vector in $\mathbb{R}^N$ via

$$\mathbf{1}_{\Gamma_a}(I) = \begin{cases} 1, & I \in \Gamma_a, \\ 0, & I \notin \Gamma_a. \end{cases}$$

Relating RatioCut to the graph Laplacian

$$\mathrm{cut}(\Gamma_a, \Gamma_a^c) = \left\langle \boldsymbol{L},\ \mathbf{1}_{\Gamma_a}\mathbf{1}_{\Gamma_a}^\top \right\rangle = \mathbf{1}_{\Gamma_a}^\top \boldsymbol{L} \mathbf{1}_{\Gamma_a}$$

$$\mathrm{RatioCut}(\{\Gamma_a\}_{a=1}^k) = \sum_{a=1}^k \frac{1}{|\Gamma_a|} \left\langle \boldsymbol{L},\ \mathbf{1}_{\Gamma_a}\mathbf{1}_{\Gamma_a}^\top \right\rangle = \langle \boldsymbol{L},\ \boldsymbol{X} \rangle ,$$

$$\underbrace{\boldsymbol{X} := \sum_{a=1}^k \frac{1}{|\Gamma_a|} \mathbf{1}_{\Gamma_a}\mathbf{1}_{\Gamma_a}^\top}_{\text{a block-diagonal projection matrix}}$$

Convex relaxation of RatioCut

Question: What constraints does $\boldsymbol{X}$ satisfy for any given $\{\Gamma_a\}_{a=1}^k$?

$$\boldsymbol{X} = \Pi \begin{bmatrix} \frac{1}{n_1}\boldsymbol{J}_{n_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_2}\boldsymbol{J}_{n_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{1}{n_k}\boldsymbol{J}_{n_k} \end{bmatrix} \Pi^\top$$

where Π is a permutation matrix, and $\boldsymbol{J}_n$ is an $n \times n$ "1" matrix.

Convex sets

Given a partition $\{\Gamma_a\}_{a=1}^k$, the corresponding $\boldsymbol{X}$ satisfies

- $\boldsymbol{X}$ is positive semidefinite, $\boldsymbol{X} \succeq 0$
- $\boldsymbol{X}$ is nonnegative, $\boldsymbol{X} \geq 0$ entrywise
- the constant vector is an eigenvector of $\boldsymbol{X}$: $\boldsymbol{X}\mathbf{1}_N = \mathbf{1}_N$
- the trace of $\boldsymbol{X}$ equals k, i.e., $\mathrm{Tr}(\boldsymbol{X}) = k$

Relaxation of RatioCut

RatioCut-SDP - SDP relaxation of RatioCut

We relax the ratio cut by

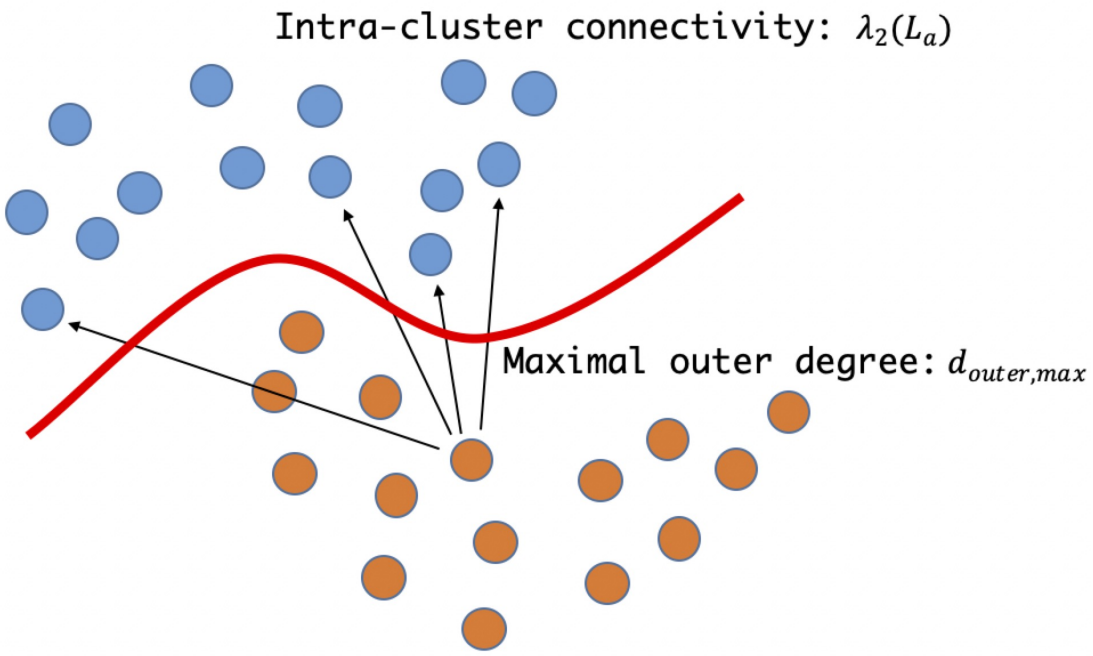
$$\min_{\boldsymbol{Z}\in\mathbb{R}^{N\times N}} \langle \boldsymbol{L}, \boldsymbol{Z}\rangle \text{ s.t. } \boldsymbol{Z} \succeq 0,\ \boldsymbol{Z} \geq 0,\ \mathrm{Tr}(\boldsymbol{Z}) = k,\ \boldsymbol{Z}\mathbf{1}_N = \mathbf{1}_N.$$

Advantage: It is a semidefinite program[2] which is solvable in polynomial time.

Question: When does convex relaxation give exact recovery?

[2] [Peng, Wei, 07], [Mixon, etc, 16], [Xing, Jordan, 03] ...

Intuition



- Intra-cluster: algebraic connectivity

$$\lambda_2(\boldsymbol{L}_a),$$

the second smallest eigenvalue of the Laplacian associated with the a-th cluster.

- Inter-cluster connectivity: the maximal outer-cluster degree.

$$d_{\text{outer,max}} = \max_i \sum_{j\text{: not share membership with node } i} w_{ij}$$

- Note that both $\lambda_2(\boldsymbol{L}_a)$ and $d_{\text{outer,max}}$ are determined by $\{\Gamma_a\}_{a=1}^k$.

Finding the optimal graph cut via SDP relaxation

Main theorem [Ling, Strohmer, FOCM]

- If a graph cut $\{\Gamma_a\}_{a=1}^k$ satisfies

$$d_{\text{outer,max}} < \frac{1}{4} \min_{1 \le a \le k} \lambda_2(\boldsymbol{L}_a)$$

  then it is the globally **optimal** graph cut under Ratiocut.
- The RatioCut convex relaxation recovers $\boldsymbol{X}$ associated to $\{\Gamma_l\}_{l=1}^k$ **exactly**!
- Even though finding the optimal Ratiocut is NP-hard, there is a regime where a poly-time algorithm works!
- Purely deterministic and depends on the algebraic properties of Laplacian. Near-optimal.
- A similar result holds for normalized cut with an interpretation from a random walk perspective.

A short tour of the proof - Game of Cones

We are dealing with

$$\min_{\boldsymbol{Z}\in\mathbb{R}^{N\times N}} \langle \boldsymbol{L}, \boldsymbol{Z}\rangle \text{ s.t. } \boldsymbol{Z} \succeq 0,\ \boldsymbol{Z} \geq 0, \underbrace{\mathrm{Tr}(\boldsymbol{Z}) = k,\ \boldsymbol{Z}\mathbf{1}_N = \mathbf{1}_N}_{\text{Linear constraints: } \mathcal{A}(\boldsymbol{Z})=\boldsymbol{b}}.$$

- Use Lagrangian duality:

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{\lambda}) = \langle \boldsymbol{L}, \boldsymbol{Z}\rangle - \langle \mathcal{A}(\boldsymbol{Z}) - \boldsymbol{b}, \boldsymbol{\lambda}\rangle$$

  where $\boldsymbol{Z}$ is in the positive semidefinite and nonnegativity cone, denoted by $\mathcal{K}$.

- Dual program

$$\max \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle \text{ s.t. } \mathcal{A}^*(\boldsymbol{\lambda}) + \boldsymbol{L} \in \mathcal{K}^*$$

  where $\mathcal{K}^*$ is the dual cone[3].

[3] $\mathcal{K}^* := \{\boldsymbol{z} : \langle \boldsymbol{z}, \boldsymbol{x}\rangle \geq 0, \forall \boldsymbol{x} \in \mathcal{K}\}$

A short tour of the proof - Game of Cones

- Dual program

$$\max \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle \text{ s.t. } \mathcal{A}^*(\boldsymbol{\lambda}) + \boldsymbol{L} \in \mathcal{K}^*$$

  where $\mathcal{K}^*$ is the dual cone[4].
- Construct $\boldsymbol{\lambda}$ to certify the optimality of a graph cut, with help of spectral graph theory.
- Construction of $\boldsymbol{\lambda}$ is rather technical and involves methods derived by X. Li, Y. Li, S. Ling, T.S. and K. Wei in [When Do Birds of a Feather Flock Together? K-Means, Proximity, and Conic Programming, Mathematical Programming, 2018.]

[4] $\mathcal{K}^* := \{\boldsymbol{z} : \langle \boldsymbol{z}, \boldsymbol{x} \rangle \geq 0, \forall \boldsymbol{x} \in \mathcal{K}\}$

Some observations

Known theoretical bounds for SDP relaxation of k-means clustering have the undesirable property that they do depend on the number of clusters.

This is not the case for our SDP relaxation of spectral clustering. The bounds are independent of the number of clusters

Our theorem also yields a certificate of optimality: Assume someone gives us a graph partition $\{\Gamma_a\}_{a=1}^k$ and claims it is the optimal RatioCut. This is in general very difficult to verify. Easy test: If the partition $\{\Gamma_a\}_{a=1}^k$ satisfies

$$d_{\text{outer,max}} < \frac{1}{4} \min_{1 \le a \le k} \lambda_2(\boldsymbol{L}_a),$$

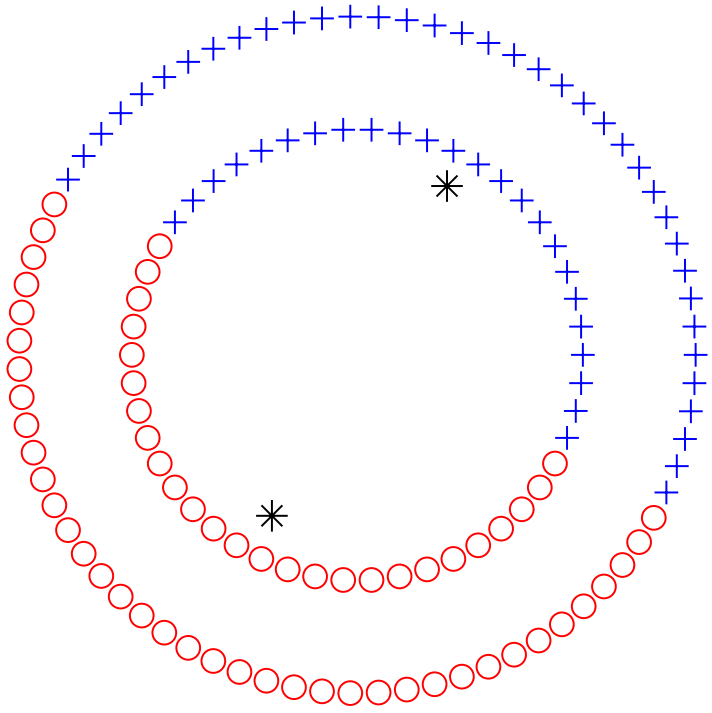
then it is indeed optimal.

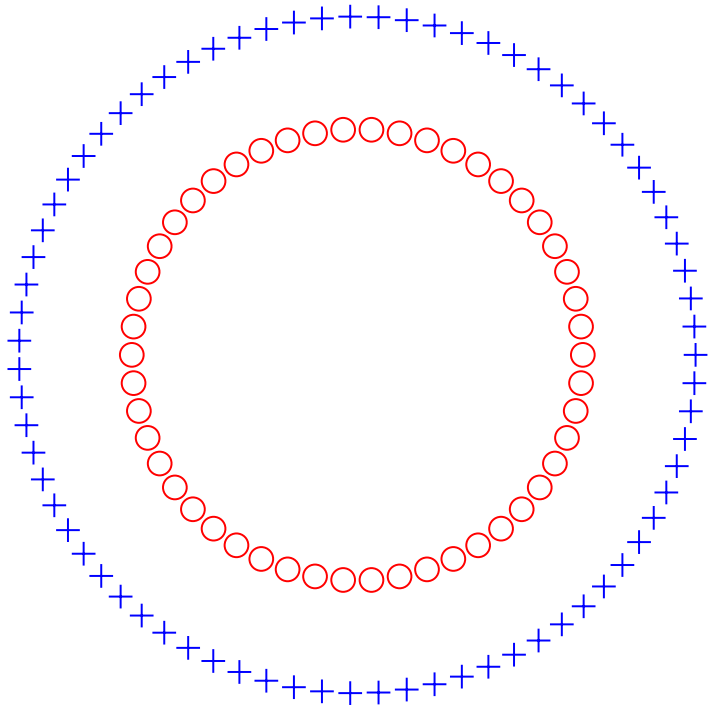
Spectral clustering for two concentric circles

We consider

$$\boldsymbol{x}_{1,i} = \begin{bmatrix} \cos(\frac{2\pi i}{n}) \\ \sin(\frac{2\pi i}{n}) \end{bmatrix}, \; 1 \leq i \leq n; \qquad \boldsymbol{x}_{2,j} = \frac{m}{n} \begin{bmatrix} \cos(\frac{2\pi j}{m}) \\ \sin(\frac{2\pi j}{m}) \end{bmatrix}, \; 1 \leq j \leq m$$

where $m > n$.





Spectral clustering for two concentric circles

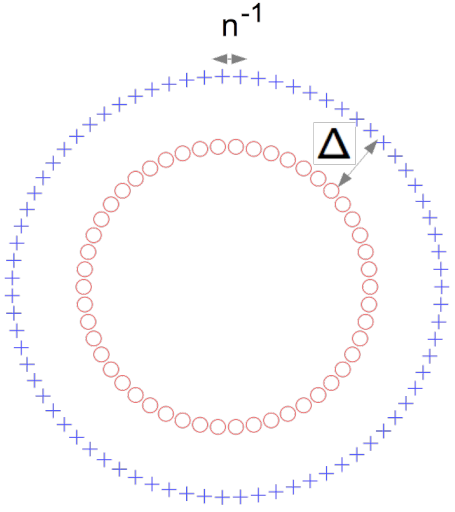
Corollary (Ling, Strohmer, 2018)

The RatioCut-SDP recovers the underlying two clusters exactly if

$$\underbrace{\Delta}_{\text{minimal separation}} = \Omega(n^{-1}).$$

where n^{-1} is the distance of two adjacent points on one circle.

It is near-optimal.

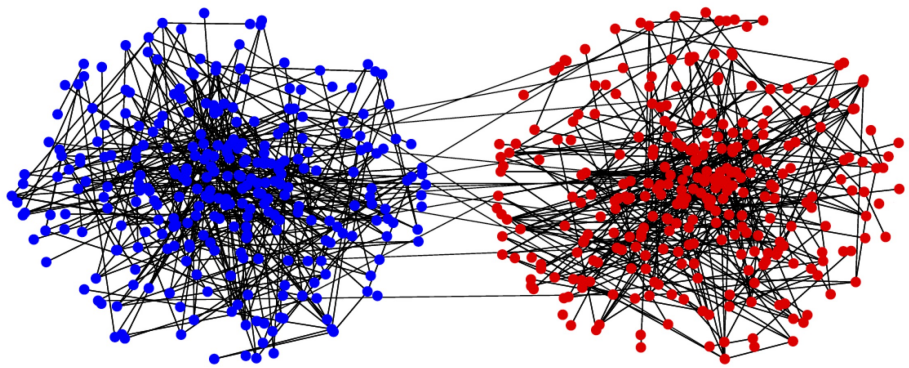


Community detection under stochastic block model

Each community has $n/2$ nodes

- if member i and j are in the same community, pair them with probability p;
- if member i and j are in different communities, pair them with probability q

Assume $\mathbf{p} > \mathbf{q}$. Given the adjacency matrix, how to find out the underlying communities?



p q p

Graph cuts and the stochastic block model

Corollary (Ling, Strohmer, 2018)

Let $p = \frac{\alpha \log n}{n}$ and $q = \frac{\beta \log n}{n}$. The RatioCut-SDP recovers the underlying communities exactly if

$$\alpha > 13\left(\sqrt{\beta} + \sqrt{2}\right)^2$$

with high probability.

The information theoretic lower bound for exact recovery[5] is

$$\alpha > (\sqrt{\beta} + \sqrt{2})^2.$$

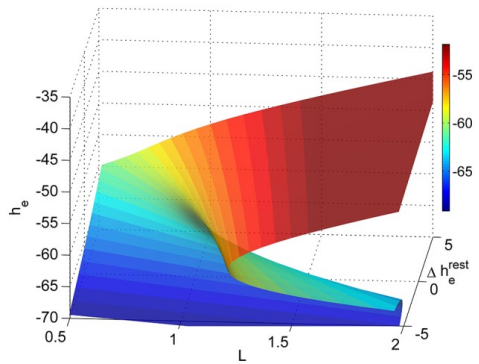
If $\alpha < (\sqrt{\beta} + \sqrt{2})^2$, exact recovery is impossible.

[5] [Abbe, Bandeira, Hall, 2016]

Outlook for random data model

Open problem

Suppose there are n data points drawn from a probability density function $p(x)$ supported on a manifold $\mathcal{M}$.
How can we estimate the second smallest eigenvalue of the graph Laplacian (either normalized or unnormalized) given the kernel function Φ and σ?



The solution will require tools from empirical process, differential geometry, spectral graph theory, etc[6].

[6] [Singer, 06], [Belkin, etc, 08], [Trillos, etc, 16], ...

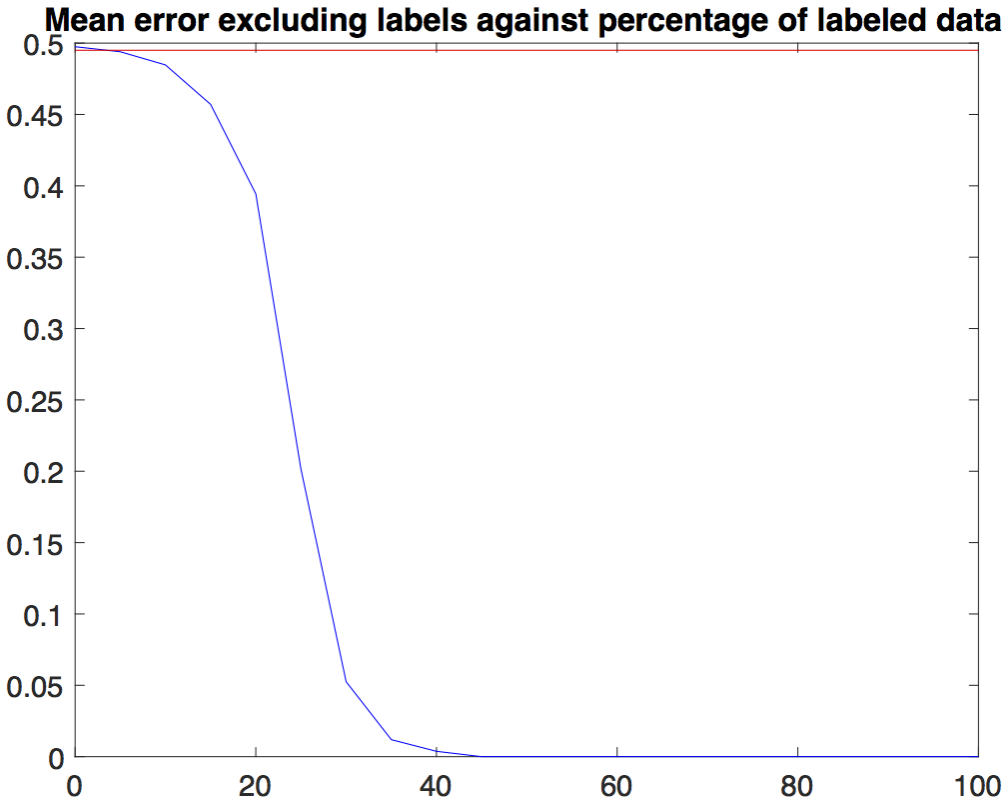
Semisupervised clustering

Assume we know for a small number of points to which cluster they belong. How can we incorporate this knowledge?

The SDP approach provides a natural framework for clustering when a few labels are known.
SDP with additional linear constraints.

$$\begin{aligned} \min \langle \boldsymbol{L}, \boldsymbol{Z} \rangle \text{ subject to } & \boldsymbol{Z} \succeq 0, \\ & \boldsymbol{Z} \geq 0, \\ & \mathrm{Tr}(\boldsymbol{Z}) = k, \\ & \boldsymbol{Z}\mathbf{1}_N = \mathbf{1}_N \\ & \boldsymbol{Z}_{j,k} = \boldsymbol{Z}_{m,n} \quad j,k,m,n \in \mathcal{I} \end{aligned}$$

Semisupervised clustering



No good theory yet

Graph Laplacians and t-SNE

(Borrowing from slides by Stefan Steinerberger)

t-SNE:
Main problem: given a set of high-dimensional points $\{x_1, \dots, x_n\} \in \mathbb{R}^d$ we would like to get an 'equivalent' representation $\{y_1, \dots, y_n\} \in \mathbb{R}^2$ so we can have a look. Somewhat ill-posed but 'clusters should remain clusters.'

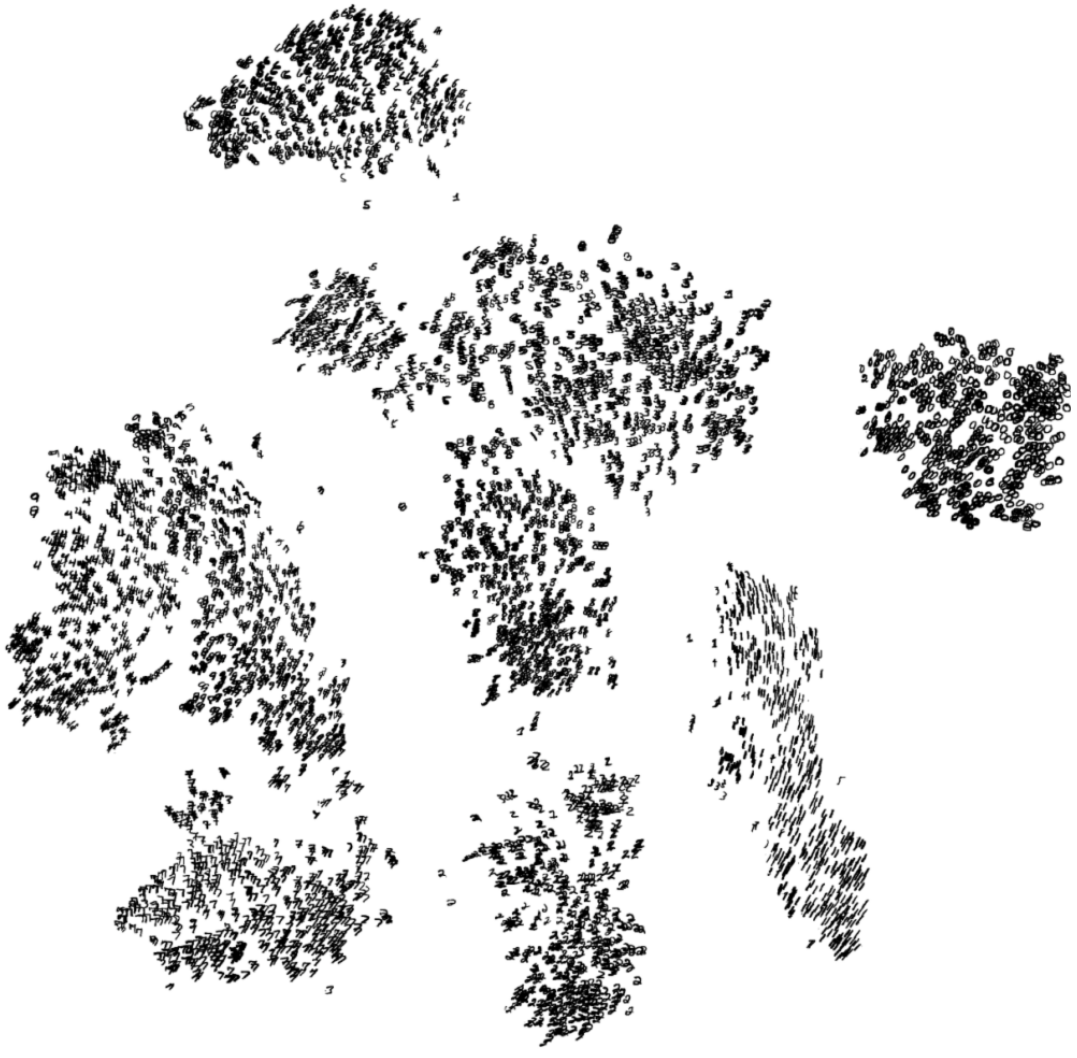
Answer: **t-distributed stochastic neighborhood embedding** (van der Maaten & Hinton, 2008)

Main idea: Turn configurations of points into probability distributions and force these distributions to be similar.

MNIST

Laplacian eigenmaps

t-SNE



Main idea of t-SNE

- Create a probability distribution in high dimensions.
- Create another on a set of points in two dimensions.
- Measure the KL-divergence between them and move the points in low dimensions around so that KL becomes small.

KL Divergence (Kullback & Leibler, 1951)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

t-SNE algorithm

Given a set of $\{x_1, \ldots, x_n\} \in \mathbb{R}^d$, t-SNE searches for $\{y_1, \ldots, y_n\} \in \mathbb{R}^2$ that minimizes the loss

$$L(y_1, \ldots, y_n) = -\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} p(x_i, x_j) \log \frac{q(y_i, y_j)}{p(x_i, x_j)},$$

where the functions p and q are given by

$$p(x_i, x_j) = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{\ell \neq i} \exp(-\|x_i - x_\ell\|^2 / 2\sigma_i^2)}$$

and

$$q(y_i, y_j) = \frac{\left(1 + (\|y_i - y_j\|^2\right)^{-1}}{\sum_{\ell \neq i} \left(1 + (\|y_i - y_\ell|^2\right)^{-1}}$$

Connections between t-SNE and Laplacians

t-SNE uses a gradient descent algorithm to compute a solution to this non-convex problem.

- Can show: [Steinerberger, *Clustering with t-SNE, provably*] For large parameters in the algorithm that computes the t-SNE solution, one obtains Laplacian Eigenmaps
- Observation: Laplacian Eigenmaps give good initialization of t-SNE algorithm

Questions:

- Can we modify Laplacian Eigenmaps by somehow adding "repulsion" to obtain better embedding in $\mathbb{R}^2$ without taking the t-SNE detour?
- Discard Kullback-Leibler interpretation of t-SNE, instead think of it as particle systems. Adopting this viewpoint, can we design good visualization methods based on particle systems & dynamical systems?

Conclusion and Outlook

- First meaningful theory for performance of spectral clustering
- Provides deterministic bounds for optimal graph cuts

Conclusion and Outlook

- First meaningful theory for performance of spectral clustering
- Provides deterministic bounds for optimal graph cuts
- Solved: Strong consistency of Laplacians for community detection [Shaofeng Deng, Oct.2019]
- Open problem: optimal estimates for random data models
- Open problem: theory for semisupervised clustering
- Open problem: how to modify Laplacian eigenmap to obtain better visualization tool

Conclusion and Outlook

- First meaningful theory for performance of spectral clustering
- Provides deterministic bounds for optimal graph cuts
- Solved: Strong consistency of Laplacians for community detection [Shaofeng Deng, Oct.2019]
- Open problem: optimal estimates for random data models
- Open problem: theory for semisupervised clustering
- Open problem: how to modify Laplacian eigenmap to obtain better visualization tool

S.Ling and T.Strohmer. Certifying Global Optimality of Graph Cuts via Semidefinite Relaxation: A Performance Guarantee for Spectral Clustering. *Foundations of Comp. Math.*, to appear. [and on the arxiv]

Interested in data science? Check out **cedar.ucdavis.edu**
Center for Data Science and Artificial Intelligence Research