

Spectral Clustering

Zitao Liu

Agenda

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Random Walk Point of View
- Perturbation Theory Point of View
- Practical Details

- **Brief Clustering Review**
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Random Walk Point of View
- Perturbation Theory Point of View
- Practical Details

CLUSTERING REVIEW

Clustering

Groups together “similar” instances in the data sample

Basic clustering problem:

- distribute data into k different groups such that data points similar to each other are in the same group
- Similarity between data points is defined in terms of some distance metric (can be chosen)

Clustering is useful for:

- **Similarity/Dissimilarity analysis**
Analyze what data points in the sample are close to each other
- **Dimensionality reduction**
High dimensional data replaced with a group (cluster) label

K-MEANS CLUSTERING

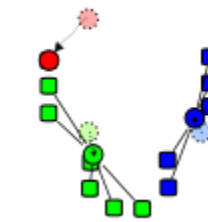
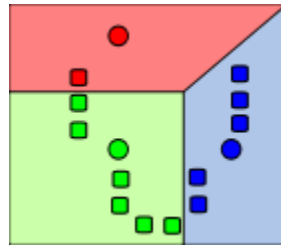
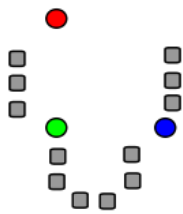
- Description

Given a set of observations $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$, where each observation is a d -dimensional real vector, k -means clustering aims to partition the n observations into k sets $(k \leq n)$ $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$\arg \min_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - u_i\|^2$$

where μ_i is the mean of points in S_i .

- Standard Algorithm



1) k initial "means" (in this case $k=3$) are randomly selected from the data set.

2) k clusters are created by associating every observation with the nearest mean.

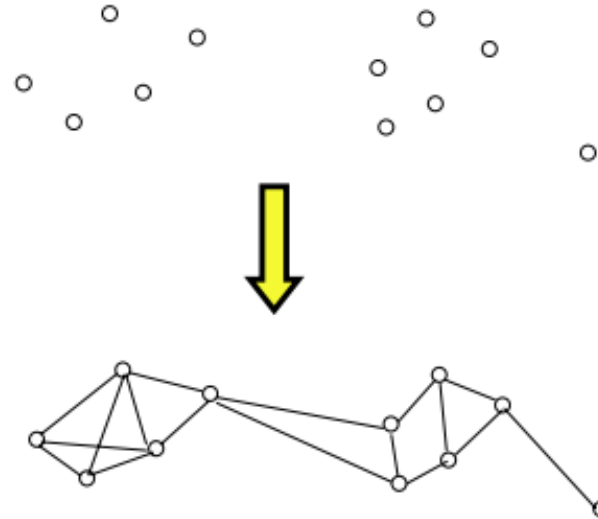
3) The centroid of each of the k clusters becomes the new means.

4) Steps 2 and 3 are repeated until convergence has been reached.

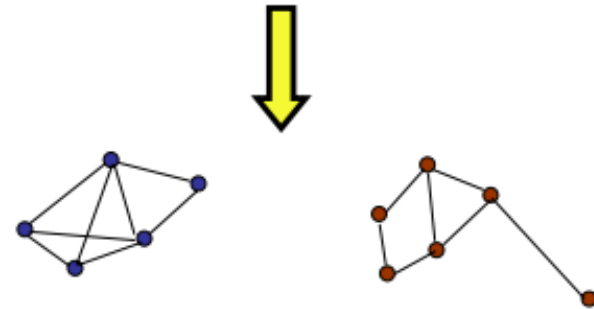
- Brief Clustering Review
- **Similarity Graph**
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Random Walk Point of View
- Perturbation Theory Point of View
- Practical Details

GENERAL

First - graph representation of data
(largely, application dependent)



Then - graph partitioning



Disconnected
graph components

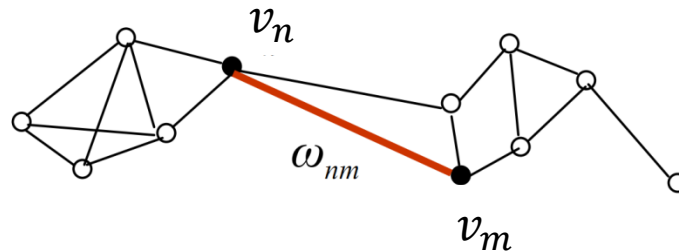


Groups of points (Weakly connections in between components
Strongly connections within components)

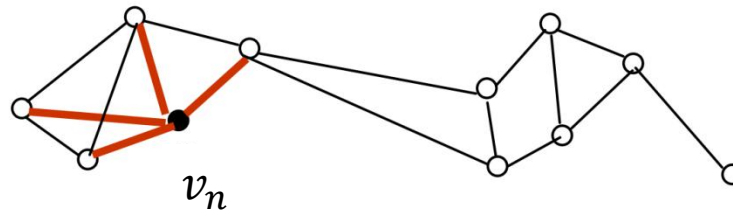
GRAPH NOTATION

$G=(V,E)$:

- Vertex set $V = \{v_1, \dots, v_n\}$
- Weighted adjacency matrix $W = (w_{ij}) \ i, j = 1, \dots, n \ w_{ij} \geq 0$



- Degree $d_i = \sum_{j=1}^n w_{ij}$



- Degree matrix Diagonal matrix with the degrees d_1, \dots, d_n on the diagonal.

GRAPH NOTATION

$G=(V,E)$:

- Indicator Vector $\mathbb{1}_A = (f_1, \dots, f_n)' \in \mathbb{R}^n$ $f_i \in \{0,1\}$
- “Size” of a subset $A \subset V$

$|A|$:= the number of vertices in A

$$vol(A) := \sum_{i \in A} d_i$$



- **Connected** A subset A 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 .
- **Connected Component** it is connected and if there are no connections between vertices in A and \bar{A} . The nonempty sets A_1, \dots, A_k form a partition of the graph if $A_i \cap A_j = \emptyset$ and $A_1 \cup \dots \cup A_k = V$.

SIMILARITY GRAPH

- ε -neighborhood graph

Connect all points whose pairwise distances are smaller than ε

- k -nearest neighbor graph

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

- fully connected graph

Connect all points with positive similarity with each other

All the above graphs are regularly used in spectral clustering!

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- **Graph Laplacian**
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Random Walk Point of View
- Perturbation Theory Point of View
- Practical Details

GRAPH LAPLACIANS

- Unnormalized Graph Laplacian

$$d_i = \sum_{j=1}^n w_{ij}$$

$$L = D - W$$

Proposition 1 (Properties of L) The matrix L satisfies the following properties:

1. For every $f \in \mathbb{R}^n$ we have

$$f'Lf = \frac{1}{2} \sum_{i,j=1}^n w_{ij}^2 (f_i - f_j)^2$$

$$\begin{aligned} f'Lf &= f'Df - f'Wf = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n f_i f_j w_{ij} \\ &= \frac{1}{2} \left(\sum_{i=1}^n d_i f_i^2 - 2 \sum_{i,j=1}^n f_i f_j w_{ij} + \sum_{j=1}^n d_j f_j^2 \right) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \end{aligned}$$

GRAPH LAPLACIANS

- Unnormalized Graph Laplacian

$$L = D - W$$

Proposition 1 (Properties of L) The matrix L satisfies the following properties:

1. For every $f \in \mathbb{R}^n$ we have

$$f'Lf = \frac{1}{2} \sum_{i,j=1}^n w_{ij}^2 (f_i - f_j)^2$$

2. L is symmetric and positive semi-definite.
3. The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$
4. L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

GRAPH LAPLACIANS

- Unnormalized Graph Laplacian

$$L = D - W$$

Proposition 2 (Number of connected components and the spectrum of L) Let G be an undirected graph with non-negative weights. The multiplicity k of the eigenvalue 0 of L equals the number of connected components A_1, \dots, A_k in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_k}$ of those components.

Proof:

When $k = 1$, a graph consisting of only one connected component we thus only have the constant one vector $\mathbb{1}$ as eigenvector with eigenvalue 0, which obviously is the indicator vector of the connected component.

When $k > 1$, L can be written in a block form. the spectrum of L is given by the union of the spectra of L_i , and the corresponding eigenvectors of L are the eigenvectors of L_i , filled with 0 at the positions of the other blocks.

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \cdots & \\ & & & L_k \end{pmatrix}$$

GRAPH LAPLACIANS

- Normalized Graph Laplacian

$$L_{sym} := D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$$

$$L_{rw} := D^{-1}L = I - D^{-1}W$$

We denote the first matrix by L_{sym} as it is a symmetric matrix, and the second one by L_{rw} as it is closely related to a random walk.

GRAPH LAPLACIANS

- Normalized Graph Laplacian

$$L_{sym} := D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$$

$$L_{rw} := D^{-1}L = I - D^{-1}W$$

Proposition 3 (Properties of L_{sym} and L_{rw}) The normalized Laplacians satisfy the following properties:

1. For every $f \in \mathbb{R}^n$ we have $f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^n w_{ij}^2 \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$
2. λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2}u$.
3. λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigen problem $Lu = \lambda Du$.
4. 0 is an eigenvalue of L_{rw} with the constant one vector $\mathbb{1}$ as eigenvector. 0 is an eigenvalue of L_{sym} with eigenvector $D^{1/2}\mathbb{1}$.
5. L_{sym} and L_{rw} are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

GRAPH LAPLACIANS

- Normalized Graph Laplacian

$$L_{sym} := D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$$

$$L_{rw} := D^{-1}L = I - D^{-1}W$$

Proposition 4 (Number of connected components and spectra of L_{sym} and 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, \dots, A_k in the graph. For L_{rw} the eigenspace of 0 is spanned by the indicator vectors $\mathbb{1}_{A_i}$ of those components. For L_{sym} , the eigenspace of 0 is spanned by the vectors $D^{1/2}\mathbb{1}_{A_i}$.

Proof. The proof is analogous to the one of Proposition 2, using Proposition 3.

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- **Spectral Clustering Algorithm**
- Graph Cut Point of View
- Random Walk Point of View
- Perturbation Theory Point of View
- Practical Details

ALGORITHM

Main trick is to change the representation of the abstract data points

x_i to points $y_i \in \mathfrak{R}^k$

1. Unnormalized Spectral Clustering
2. Normalized Spectral Clustering 1
3. Normalized Spectral Clustering 2

ALGORITHM

- Unnormalized Graph Laplacian

$$L = D - W$$

Unnormalized spectral clustering

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

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

Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.

ALGORITHM

- Normalized Graph Laplacian

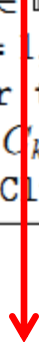
$$L_{rw} := D^{-1}L = I - D^{-1}W$$

Normalized spectral clustering according to Shi and Malik (2000)

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

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

Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.



Proposition 3 (Properties of L_{sym} and L_{rw}) *The normalized Laplacians satisfy the following properties:*

3. λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigenproblem $Lu = \lambda Du$.

ALGORITHM

- Normalized Graph Laplacian

$$L_{\text{sym}} := D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$$

Normalized spectral clustering according to Ng, Jordan, and Weiss (2002)

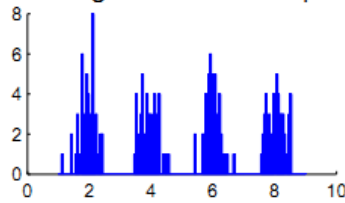
Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

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

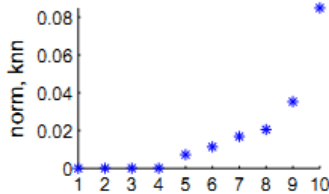
Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.

ALGORITHM

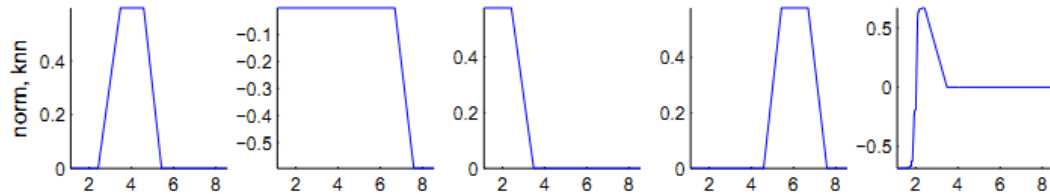
Histogram of the sample



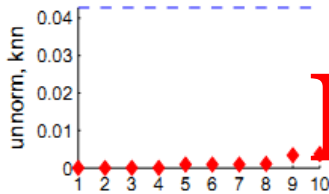
Eigenvalues



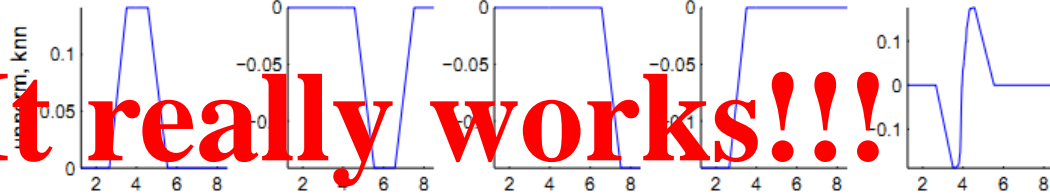
Eigenvector 1 Eigenvector 2 Eigenvector 3 Eigenvector 4 Eigenvector 5



Eigenvalues

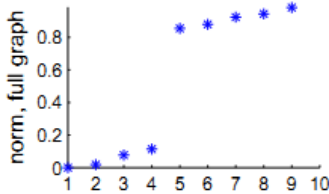


Eigenvector 1 Eigenvector 2 Eigenvector 3 Eigenvector 4 Eigenvector 5

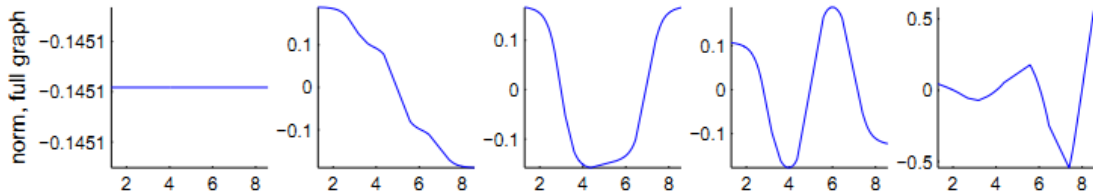


It really works!!!

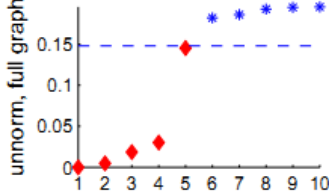
Eigenvalues



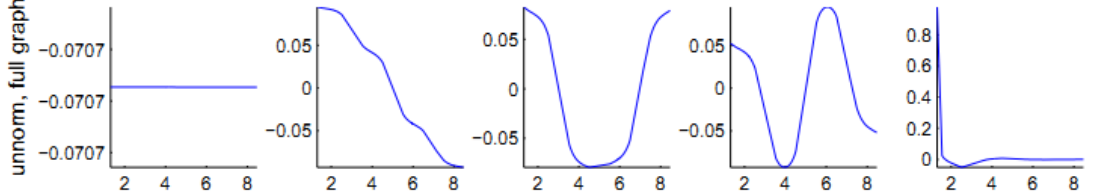
Eigenvector 1 Eigenvector 2 Eigenvector 3 Eigenvector 4 Eigenvector 5



Eigenvalues



Eigenvector 1 Eigenvector 2 Eigenvector 3 Eigenvector 4 Eigenvector 5



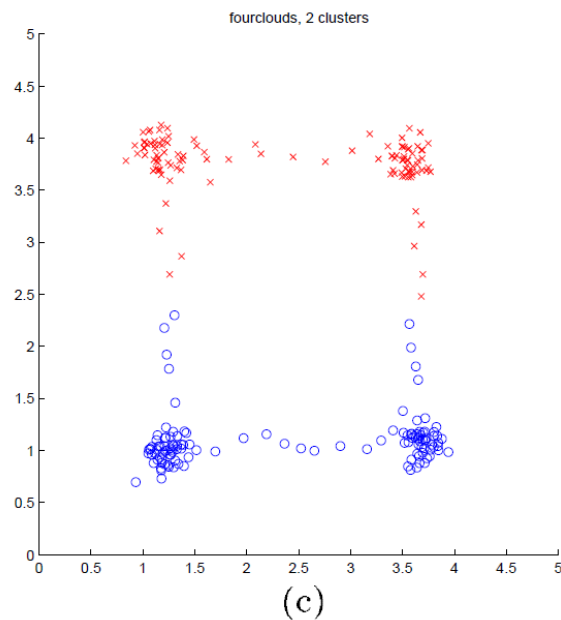
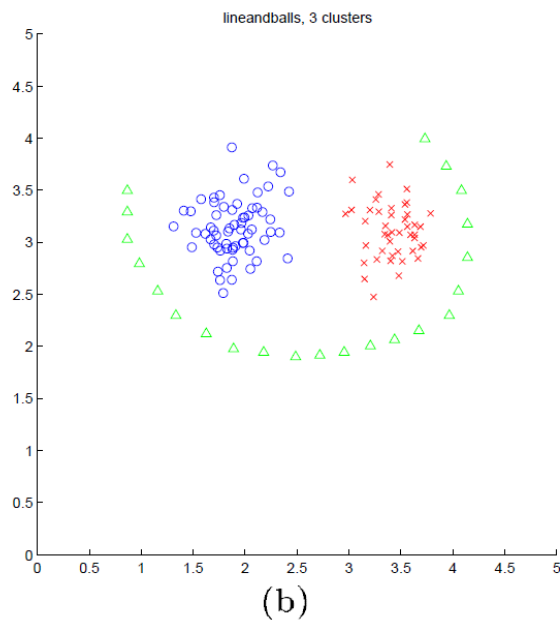
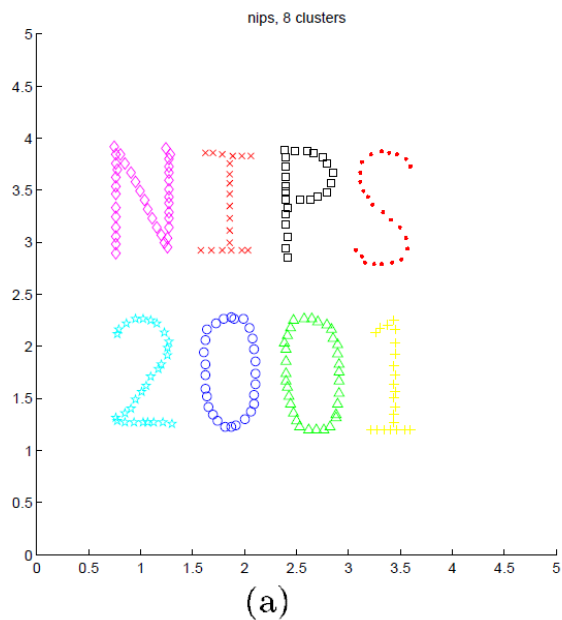
ALGORITHM

On Spectral Clustering: Analysis and an algorithm

Andrew Y. Ng
CS Division
U.C. Berkeley
ang@cs.berkeley.edu

Michael I. Jordan
CS Div. & Dept. of Stat.
U.C. Berkeley
jordan@cs.berkeley.edu

Yair Weiss
School of CS & Engr.
The Hebrew Univ.
yweiss@cs.huji.ac.il

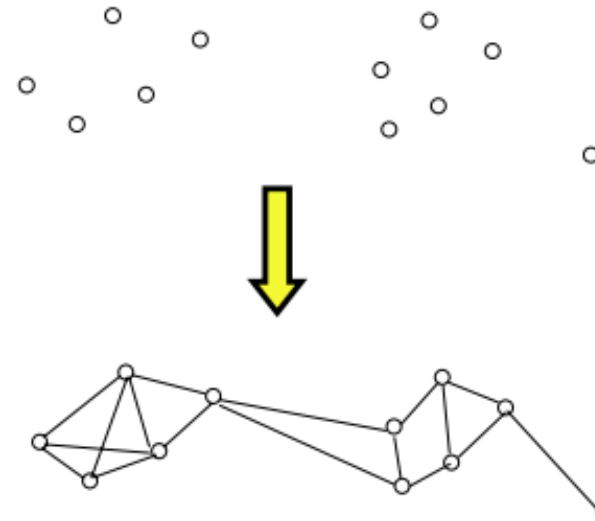


Spectral Clustering

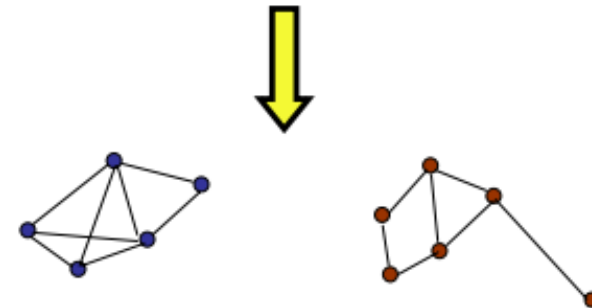
- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- **Graph Cut Point of View**
- Random Walk Point of View
- Perturbation Theory Point of View
- Practical Details

GRAPH CUT

First - graph representation of data
(largely, application dependent)



Then - graph partitioning



Disconnected
graph components



Groups of points (Weakly connections in between components
Strongly connections within components)

GRAPH CUT

$G=(V,E)$:

- For two not necessarily disjoint set $A, B \subset V$, we define

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

- Minicut: choosing a partition A_1, A_2, \dots, A_K which minimizes

$$cut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k W(A_i, \bar{A}_i)$$

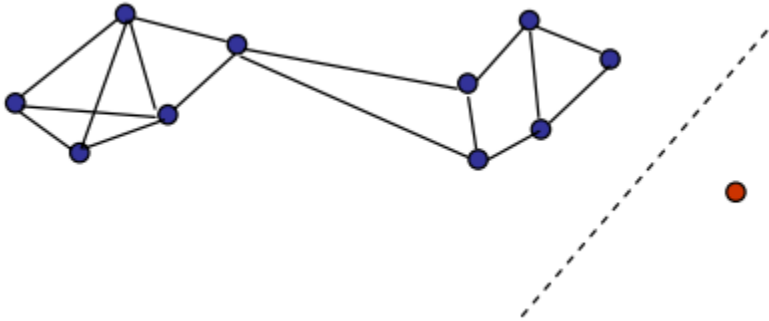
Cut between 2 sets $cut(A_1, A_2) = \sum_{n \in A_1} \sum_{m \in A_2} w_{nm}$



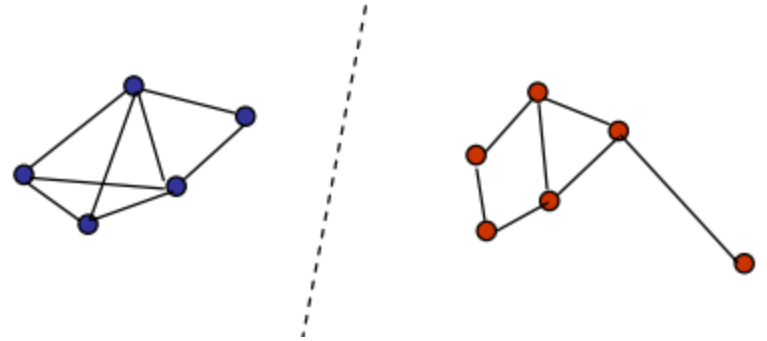
GRAPH CUT

Problems!!!

- Sensitive to outliers



What we get



What we want

GRAPH CUT

Solutions

$|A|$:= the number of vertices in A

$$vol(A) := \sum_{i \in A} d_i$$

- RatioCut(Hagen and Kahng, 1992)

$$RatioCut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{|A_i|}$$

- Ncut(Shi and Malik, 2000)

$$Ncut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{vol(A_i)} = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{vol(A_i)}$$

GRAPH CUT

Problem!!!

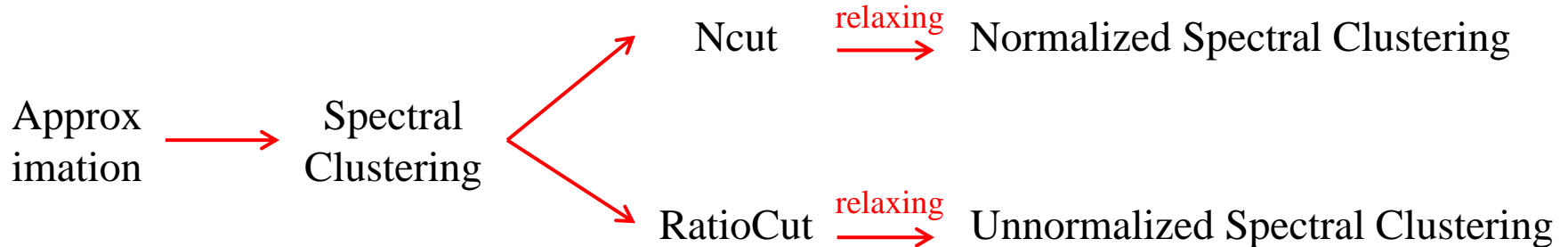
- NP hard

Solution!!!

- Approximation

$$\text{RatioCut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

$$\text{Ncut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}$$



GRAPH CUT

- Approximation RatioCut for $k=2$

Our goal is to solve the optimization problem:

$$\min_{A \subset V} \text{RatioCut}(A, \bar{A})$$

Rewrite the problem in a more convenient form:

Given a subset $A \subset V$, we define the vector $f = (f_1, \dots, f_n)' \in \mathbb{R}^n$ with entries

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|}, & \text{if } v_i \in A \\ -\sqrt{|\bar{A}|/|A|}, & \text{if } v_i \in \bar{A} \end{cases}$$

Magic happens!!!

GRAPH CUT

- Approximation RatioCut for $k=2$

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|}, & \text{if } v_i \in A \\ -\sqrt{|\bar{A}|/|A|}, & \text{if } v_i \in \bar{A} \end{cases}$$

$$\text{RatioCut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

$$\text{cut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k W(A_i, \bar{A}_i) \quad W(A, B) := \sum_{i \in A, j \in B} w_{ij}$$

$$f' L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$$

$$= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} \left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2$$

$$= \text{cut}(A, \bar{A}) \left(\frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2 \right)$$

$$= \text{cut}(A, \bar{A}) \left(\frac{|A| + |\bar{A}|}{|A|} + \frac{|A| + |\bar{A}|}{|\bar{A}|} \right)$$

$$= |V| \cdot \text{RatioCut}(A, \bar{A}).$$

GRAPH CUT

- Approximation RatioCut for $k=2$

Additionally, we have

$$\sum_{i=1}^n f_i = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}} = |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0.$$

The vector f as defined before is orthogonal to the constant one vector $\mathbb{1}$.

f satisfies

$$\|f\|^2 = \sum_{i=1}^n f_i^2 = |A| \frac{|\bar{A}|}{|A|} + |\bar{A}| \frac{|A|}{|\bar{A}|} = |\bar{A}| + |A| = n.$$

GRAPH CUT

- Approximation RatioCut for $k=2$

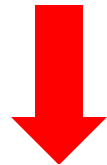
$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{if } v_i \in \bar{A}. \end{cases}$$

$$\min_{ACV} \text{RatioCut}(A, \bar{A}).$$



$$\begin{aligned} f' L f &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \\ &= |V| \cdot \text{RatioCut}(A, \bar{A}). \end{aligned}$$

$$\min_{ACV} f' L f \text{ subject to } f \perp \mathbf{1} \quad \|f\| = \sqrt{n}.$$



Relaxation !!!

$$\min_{f \in \mathbb{R}^n} f' L f \text{ subject to } f \perp \mathbf{1}, \|f\| = \sqrt{n}.$$



Rayleigh-Ritz Theorem

f is the eigenvector corresponding to the second smallest eigenvalue of L (the smallest eigenvalue of L is 0 with eigenvector $\mathbb{1}$)

GRAPH CUT

- Approximation RatioCut for $k=2$

f is the eigenvector corresponding to the second smallest eigenvalue of L

Use the sign as
indicator
function

re-convert

f_i as points in R
and do K-means

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \bar{A} & \text{if } f_i < 0. \end{cases}$$

$$\begin{cases} v_i \in A & \text{if } f_i \in C \\ v_i \in \bar{A} & \text{if } f_i \in \bar{C}. \end{cases}$$

Only works for $k = 2$


More General, works for any k


GRAPH CUT


- Approximation RatioCut for arbitrary k

Given a partition of V into k sets A_1, A_2, \dots, A_k , we define k indicator vectors $h_j = (h_{1,j}, \dots, h_{n,j})'$ by

$$h_{i,j} = \begin{cases} \frac{1}{\sqrt{|A_j|}}, & \text{if } v_i \in A_j \\ 0, & \text{otherwise} \end{cases} \quad (i=1, \dots, n; j=1, \dots, k)$$


$$h_i' L h_i = \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$


$$h_i' L h_i = (H' L H)_{ii}$$


$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k h_i' L h_i = \sum_{i=1}^k (H' L H)_{ii} = \text{Tr}(H' L H)$$

$H \in \mathbb{R}^{n \times k}$, containing those k Indicator vectors as columns. Columns in H are orthonormal to each other, that is $H' H = I$

GRAPH CUT

- Approximation RatioCut for arbitrary k

$$h_{i,j} = \begin{cases} \frac{1}{\sqrt{|A_j|}}, & \text{if } v_i \in A_j \\ 0, & \text{otherwise} \end{cases}$$

Problem reformulation:

minimizing $\text{RatioCut}(A_1, \dots, A_k)$



$\min_{A_1, \dots, A_k} \text{Tr}(H' L H)$ subject to $H' H = I$



Relaxation !!!

$\min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H' L H)$ subject to $H' H = I$



Rayleigh-Ritz Theorem

Optimal H is the first k eigenvectors of L as columns.

GRAPH CUT

- Approximation Ncut for $k=2$
$$Ncut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{vol(A_i)} = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{vol(A_i)}$$

Our goal is to solve the optimization problem:

$$\min_{A \subset V} Ncut(A, \bar{A})$$

Rewrite the problem in a more convenient form:

Given a subset $A \subset V$, we define the vector $f = (f_1, \dots, f_n)' \in \mathbb{R}^n$ with entries

$$f_i = \begin{cases} \sqrt{\frac{vol(\bar{A})}{vol(A)}} & \text{if } v_i \in A \\ -\sqrt{\frac{vol(A)}{vol(\bar{A})}} & \text{if } v_i \in \bar{A} \end{cases}$$

Similar to above one can check that:

$$(Df)' \mathbf{1} = 0, f' Df = vol(V), \text{ and } f' Lf = vol(V) Ncut(A, \bar{A})$$

GRAPH CUT

- Approximation Ncut for k=2

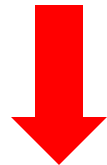
$$f_i = \begin{cases} \sqrt{\frac{\text{vol}(\bar{A})}{\text{vol} A}} & \text{if } v_i \in A \\ -\sqrt{\frac{\text{vol}(A)}{\text{vol}(\bar{A})}} & \text{if } v_i \in \bar{A}. \end{cases} \quad (6)$$

$$\min_{A \subset V} \text{Ncut}(A, \bar{A})$$



$$f'Lf = \text{vol}(V)\text{Ncut}(A, \bar{A})$$

$$\min_A f'Lf \text{ subject to } f \text{ as in (6), } Df \perp \mathbf{1}, f'Df = \text{vol}(V)$$



Relaxation !!!

$$\min_{f \in \mathbb{R}^n} f'Lf \text{ subject to } Df \perp \mathbf{1}, f'Df = \text{vol}(V)$$



Substitute $g := D^{1/2}f$

$$\min_{g \in \mathbb{R}^n} g'D^{-1/2}LD^{-1/2}g \text{ subject to } g \perp D^{\frac{1}{2}}\mathbf{1}, \quad \|g\|^2 = \text{vol}(V)$$

Rayleigh-Ritz Theorem!!!

GRAPH CUT

- Approximation Ncut for arbitrary k

$$h_{i,j} = \begin{cases} 1/\sqrt{\text{vol}(A_j)} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases}$$

Problem reformulation:

$$\min_{A \subset V} \text{Ncut}(A_1, A_2, \dots, A_k)$$



$$\min_{A_1, \dots, A_k} \text{Tr}(H' L H) \text{ subject to } H' D H = I$$



Relaxation !!!

Re-substituting $H = D^{-1/2} T$

$$\min_{T \in \mathbb{R}^{n \times k}} \text{Tr}(T' D^{-1/2} L D^{-1/2} T) \text{ subject to } T' T = I$$



Rayleigh-Ritz Theorem

T contains the first k eigenvectors of L_{sym} as columns.

Re-substituting $H = D^{-1/2} T$, solution H contains the first k eigenvectors of L_{rw} .

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- **Random Walk Point of View**
- Perturbation Theory Point of View
- Practical Details

RANDOM WALK

- A random walk on a graph is a stochastic process which randomly jumps from vertex to vertex.
- Random walk stays long within the same cluster and seldom jumps between clusters.
- A balanced partition with a low cut will also have the property that the random walk does not have many opportunities to jump between clusters.

RANDOM WALK

- Transition probability p_{ij} of jumping from v_i to v_j

$$p_{ij} = w_{ij}/d_i$$

- The transition matrix $P = (p_{ij})$ $i, j = 1, \dots, n$ of random walk is defined by

$$P = D^{-1}W$$

- If the graph is connected and non-bipartite, the random walk always processes a unique stationary distribution $\pi = (\pi_1, \dots, \pi_n)'$, where $\pi_i = d_i/\text{vol}(V)$. ($d_i = \sum_{j=1}^n w_{ij}$, $\text{vol}(V) := \sum_{i \in V} d_i$)

RANDOM WALK

- Relationship between L_{rW} and P .

$$L_{rW} = I - P$$

- λ is an eigenvalue of L_{rW} with eigenvector u if and only if $1 - \lambda$ is an eigenvalue of P with eigenvector u .
- The largest eigenvectors of P and the smallest eigenvectors of L_{rW} can be used to describe cluster properties of the graph.

RANDOM WALK

- Random walks and Ncut

Proposition 5 (Ncut via transition probabilities) Let G be connected and non bi-partite. Assume that we run the random walk $(X_t)_{t \in \mathbb{N}}$ starting with X_0 in the stationary distribution π . For disjoint subsets $A, B \subset V$, denote by $P(B|A) := P(X_1 \in B | X_0 \in A)$. Then:

$$Ncut(A, \bar{A}) = P(\bar{A}|A) + P(A|\bar{A}).$$

RANDOM WALK

- Random walks and Ncut

Proposition 5 (Ncut via transition probabilities) Let G be connected and non bipartite. Assume that we run the random walk $(X_t)_{t \in \mathbb{N}}$ starting with X_0 in the stationary distribution π . For disjoint subsets $A, B \subset V$, denote by $P(B|A) := P(X_1 \in B | X_0 \in A)$. Then:

$$\text{Ncut}(A, \bar{A}) = P(\bar{A}|A) + P(A|\bar{A}).$$

Proof. First of all observe that

$$\begin{aligned} P(X_0 \in A, X_1 \in B) &= \sum_{i \in A, j \in B} P(X_0 = i, X_1 = j) = \sum_{i \in A, j \in B} \pi_i p_{ij} \\ &= \sum_{i \in A, j \in B} \frac{d_i}{\text{vol}(V)} \frac{w_{ij}}{d_i} = \frac{1}{\text{vol}(V)} \sum_{i \in A, j \in B} w_{ij} \end{aligned}$$

Using this we obtain

$$\begin{aligned} P(X_1 \in B | X_0 \in A) &= \frac{P(X_0 \in A, X_1 \in B)}{P(X_0 \in A)} \\ &= \left(\frac{1}{\text{vol}(V)} \sum_{i \in A, j \in B} w_{ij} \right) \left(\frac{\text{vol}(A)}{\text{vol}(V)} \right)^{-1} = \frac{\sum_{i \in A, j \in B} w_{ij}}{\text{vol}(A)} \end{aligned}$$

Now the proposition follows directly with the definition of Ncut.

RANDOM WALK

- Random walks and Ncut

Proposition 5 (Ncut via transition probabilities) Let G be connected and non bi-partite. Assume that we run the random walk $(X_t)_{t \in \mathbb{N}}$ starting with X_0 in the stationary distribution π . For disjoint subsets $A, B \subset V$, denote by $P(B|A) := P(X_1 \in B | X_0 \in A)$. Then:

$$\text{Ncut}(A, \bar{A}) = P(\bar{A}|A) + P(A|\bar{A}).$$

It tells us that when minimizing Ncut, we actually look for a cut through the graph such that **A random walk seldom transitions from A to \bar{A} and vice versa.**

RANDOM WALK

- What is commute distance

The commute distance (resistance distance) c_{ij} between two vertices v_i and v_j is the expected time it takes the random walk to travel from vertex v_i to vertex v_j and back.

The commute distance between two vertices decrease if there are many different short ways to get from vertex v_i to vertex v_j .

Points which are **connected by a short path** in the graph and **lie in the same high-density region** of the graph are considered closer to each other than points which are connected by a short path but lie in different high-density regions of the graph.

Well-suited for Clustering

RANDOM WALK

- How to calculate commute distance

Generalized inverse (also called pseudo-inverse or Moore-Penrose inverse)

L can be decomposed as $L = U \Lambda U'$, and L is not invertible.

Define *generalized inverse* as $L^\dagger = U \Lambda^\dagger U'$, and Λ^\dagger is the diagonal matrix with the eigenvalues $\lambda_1, \dots, \lambda_n$ on the diagonal entries $1/\lambda_i$ if $\lambda_i \neq 0$ and 0 if $\lambda_i = 0$.

The entries of L^\dagger can be computed as $l_{ij}^\dagger = \sum_{k=2}^n \frac{1}{\lambda_k} u_{ik} u_{jk}$

RANDOM WALK

- How to calculate commute distance

Proposition 6 (Commute distance) Let $G=(V,E)$ a connected, undirected graph. Denote by c_{ij} the commute distance between vertex v_i and vertex v_j , and by $L^\dagger = (l_{ij}^\dagger)_{i,j=1,\dots,n}$ the generalized inverse of L . Then we have:

$$c_{ij} = \text{vol}(V)(l_{ii}^\dagger - 2l_{ij}^\dagger + l_{jj}^\dagger) = \text{vol}(V)(e_i - e_j)'L^\dagger(e_i - e_j)$$

$e_i = (0, \dots, 0, 1, 0, \dots, 0)'$ as the i -th unit vector.

This result has been published by Klein and Randić(1993), where it has been proved by methods of electrical network theory.

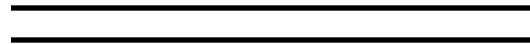
RANDOM WALK

- Proposition 6's consequence

$$c_{ij} = \text{vol}(V)(l_{ii}^\dagger - 2l_{ij}^\dagger + l_{jj}^\dagger) = \text{vol}(V)(e_i - e_j)'L^\dagger(e_i - e_j)$$

*Construct
an embedding*

Commuter Distance
between v_i and v_j



Euclidean Distance
between z_i and z_j

Choose z_i as the point in \mathbb{R}^n corresponding to the i -th row of the matrix $U(\Lambda^\dagger)^{1/2}$

$$\langle z_i, z_j \rangle = e_i' L^\dagger e_j \quad \text{and} \quad c_{ij} = \text{vol}(V) \|z_i - z_j\|^2$$

RANDOM WALK

- A loose relation between spectral clustering and commute distance.

Spectral Clustering

1. Map the vertices of the graph on the rows y_i of the matrix U
2. Only take the first k columns of the matrix

Commute Distance

1. Map the vertices on the rows z_i of the matrix $(\Lambda^\dagger)^{1/2}U$
2. Commute time embedding takes all columns

Several authors justify that spectral clustering constructs clusters based on the Euclidean distances between the y_i can be interpreted as building clusters of the vertices in the graph based on the commute distance.

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Random Walk Point of View
- **Perturbation Theory Point of View**
- Practical Details

PERTURBATION THEORY

- Perturbation theory studies the question of how eigenvalues and eigenvectors of a matrix A change if we add a small perturbation H .

$$\text{perturbed matrix } \tilde{A} := A + H$$

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 .

PERTURBATION THEORY

Theorem 7 (Davis-Kahan) *Let $A, H \in \mathbb{R}^{n \times n}$ be symmetric matrices, and let $\|\cdot\|$ be the Frobenius norm or the two-norm for matrices, respectively. Consider $\tilde{A} := A + H$ as 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 (more formally, V_1 is the image of the spectral projection induced by $\sigma_{S_1}(A)$). 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 and \tilde{V}_1 is bounded by

$$d(V_1, \tilde{V}_1) \leq \frac{\|H\|}{\delta}.$$

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

Below we will see that 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.

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Random Walk Point of View
- Perturbation Theory Point of View
- **Practical Details**

PRACTICAL DETAILS

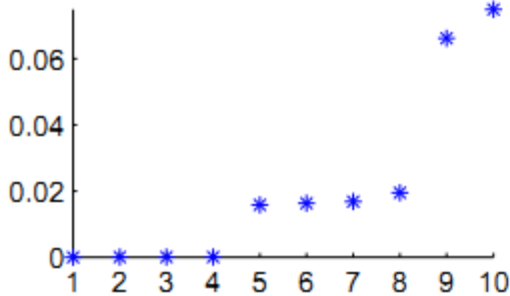
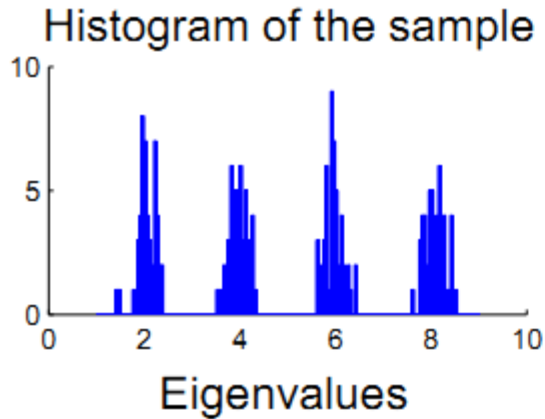
- Constructing the similarity graph
 1. **Similarity Function Itself** Make sure that points which are considered to be “very similar” by the similarity function are also closely related in the application the data comes from.
 2. **Type of Similarity Graph** Which one to choose from those three types.
General recommendation: k -nearest neighbor graph.
 3. **Parameters of Similarity Graph**(k or ε)
 1. KNN: k in order of $\log(n)$;
 2. mutual KNN: k significantly larger than standard KNN;
 3. ε -neighborhood graph: longest edge of MST;
 4. fully connected graph: σ in order of the mean distance of a point to its k -th nearest neighbor. Or choose $k = \varepsilon$.

PRACTICAL DETAILS

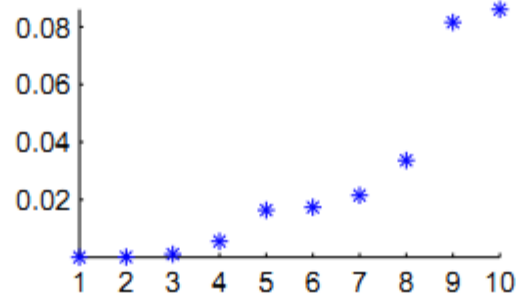
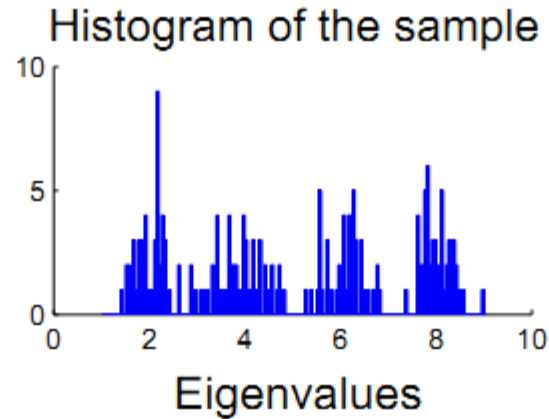
- Computing Eigenvectors
 1. How to compute the first eigenvectors efficiently for large L
 2. Numerical eigensolvers converge to some orthonormal basis of the eigenspace.

- Number of Clusters
 1. General Strategies
 2. Eigengap heuristic(Choose the number k such that all eigenvalues $\lambda_1, \dots, \lambda_k$ are very small, but λ_{k+1} is relatively large)

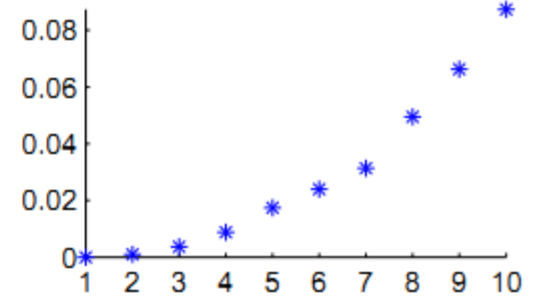
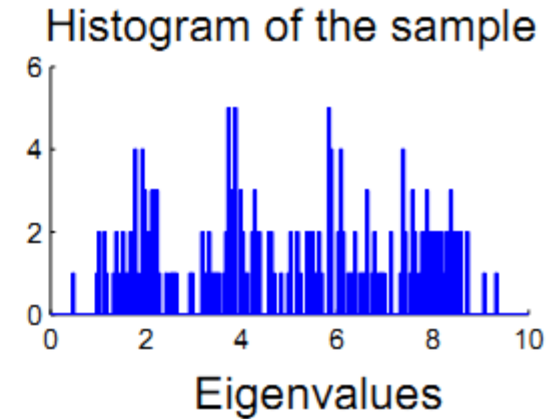
PRACTICAL DETAILS



Well Separated



More Blurry



Overlap So Much

Eigengap Heuristic usually works well if the data contains very well pronounced clusters, but in ambiguous cases it also returns ambiguous results.

PRACTICAL DETAILS

- The k-means step

It is not necessary. People also use other techniques

- Which graph Laplacian should be used?

Look at the degree distribution. There are several arguments which advocate for using normalized rather than unnormalized spectral clustering, and in the normalized case to use the eigenvectors of L_{rw} rather than those of L_{sym}

PRACTICAL DETAILS

- Which graph Laplacian should be used?

Why normalized is better than unnormalized spectral clustering?

Objective1:

1. We want to find a partition such that points in different clusters are dissimilar to each other, that is we want to minimize the between-cluster similarity. In the graph setting, this means to minimize $cut(A, \bar{A})$.

Both RatioCut and Ncut directly implement

Objective2:

2. We want to find a partition such that points in the same cluster are similar to each other, that is we want to maximize the within-cluster similarities $W(A, A)$, and $W(\bar{A}, \bar{A})$.

Only Ncut implements

Normalized spectral clustering implements both clustering objectives mentioned above, while unnormalized spectral clustering only implements the first objective.

PRACTICAL DETAILS

- Which graph Laplacian should be used?

Why the eigenvectors of L_{rw} are better than those of L_{sym} ?

1. Eigenvectors of L_{rw} are cluster indicator vectors \mathbb{I}_{A_i} , while the eigenvectors of L_{sym} are additionally multiplied with $D^{1/2}$, which might lead to undesired artifacts.
2. Using L_{sym} also does not have any computational advantages.

REFERENCE

- Ulrike Von Luxburg. A Tutorial on Spectral Clustering. Max Planck Institute for Biological Cybernetics Technical Report No. TR-149.
- Marina Meila and Jianbo Shi. A random walks view of spectral segmentation. In Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS), 2001.
- A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems (NIPS),14, 2001.
- A. Azran and Z. Ghahramani. A new Approach to Data Driven Clustering. In International Conference on Machine Learning (ICML),11, 2006.
- A. Azran and Z. Ghahramani. Spectral Methods for Automatic Multiscale Data Clustering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.
- Arik Azran. A Tutorial on Spectral Clustering.
http://videlectures.net/mlcued08_azran_mcl/

SPECTRAL CLUSTERING

Thank you