Spectral Clustering

Outline

- Brief Clustering Review
- Graph Terminology
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Practical Details

- Brief Clustering Review
- Graph Terminology
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Practical Details

What is clustering?

- Given:
  - Data set of objects
  - Relationships between these objects

- Goal: Find meaningful groups of objects s.t.
  - Objects in the same group are “similar”
  - Objects in different group are “dissimilar”

K-MEANS CLUSTERING

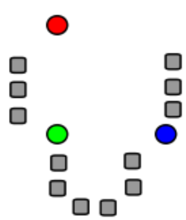
- Description

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

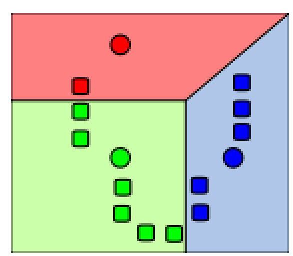
$$\underset{S}{\arg\min} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - u_i \right\|^2$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

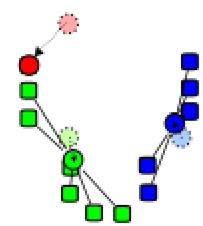
- Standard Algorithm



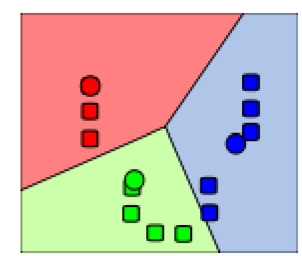
1) k initial "means" (in this case k=3) are randomly selected from the data set.



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

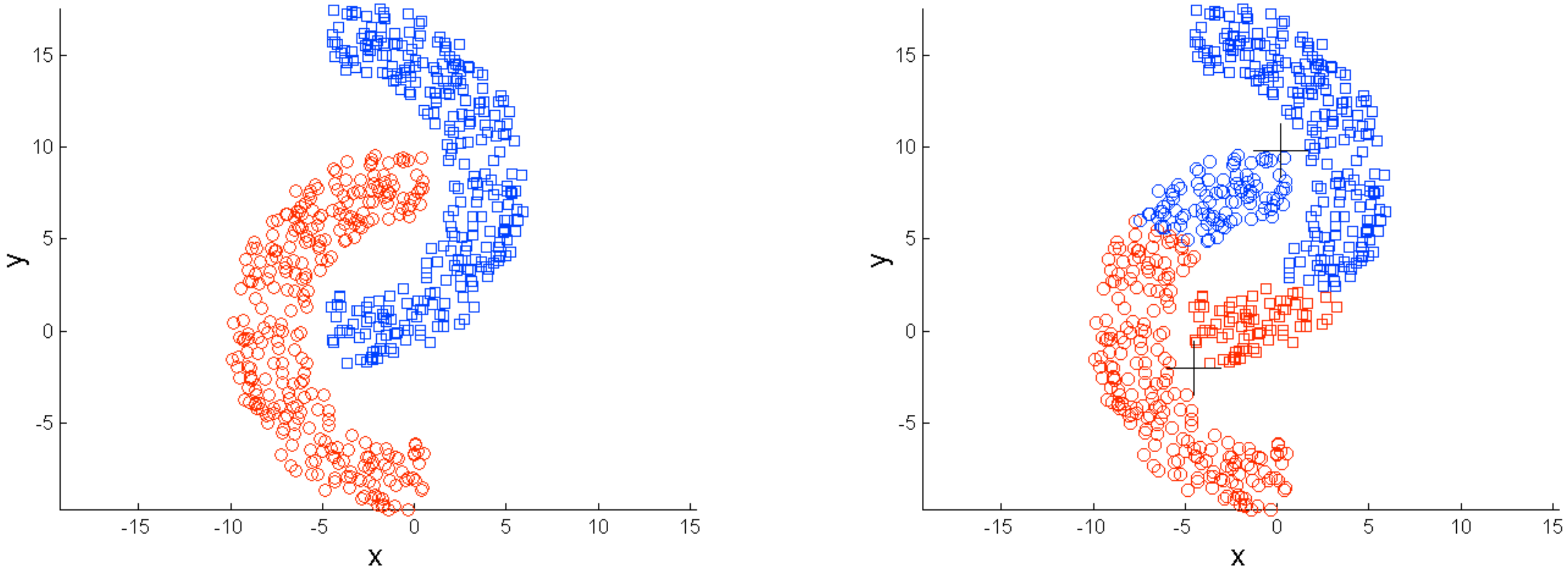


3) The centroid of each of the k clusters becomes the new means.



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

Data Manifold



- Distance based method (like k-means) may not work well
- Must uncover the manifold of data

SPECTRAL CLUSTERING

- Obtain data representation in the low-dimensional space that can be easily clustered

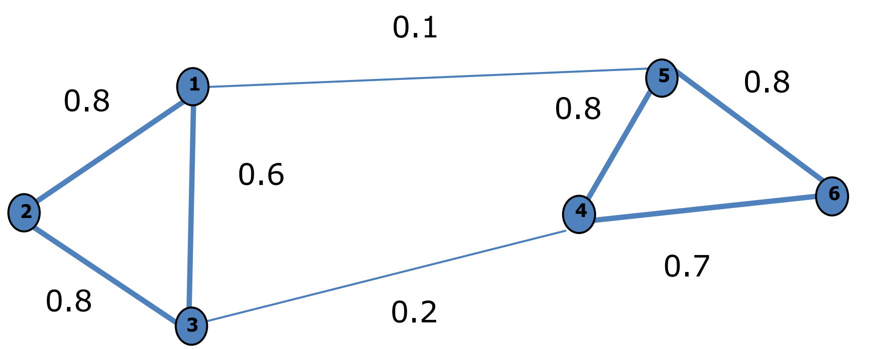
- Cluster points using eigenvectors of matrices derived from the data

- Brief Clustering Review
- Graph Terminology
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Practical Details

GRAPH NOTATION: Similarity Matrix

$G=(V,E)$:

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



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

- Important properties: symmetric matrix
  - Eigenvalues are real
  - Eigenvector could span orthogonal base

GRAPH NOTATION: Similarity Matrix

- 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

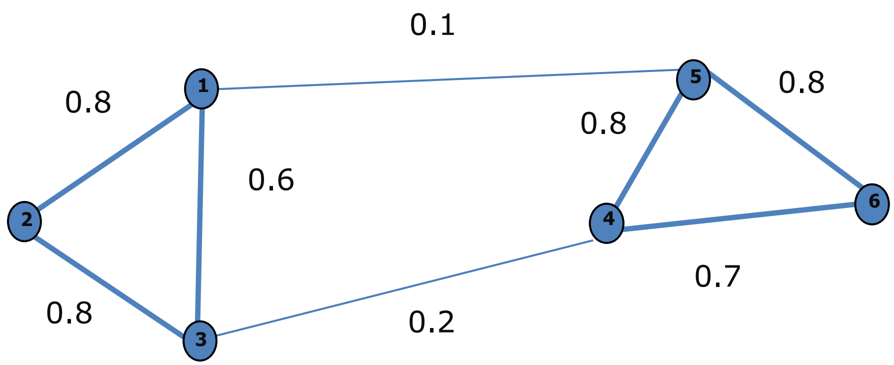
  Connect all points with positive similarity with each other

All the above graphs are regularly used in spectral clustering!

GRAPH NOTATION: Degree

$G=(V,E)$:

- Vertex set $V = \{v_1, \ldots, v_n\}$
- Degree: total weight of edges incident to vertex i. $d_i = \sum_{j=1}^{n} w_{ij}$



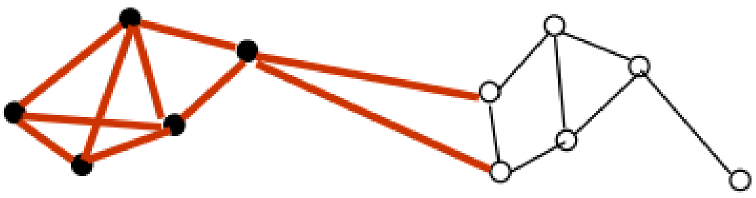
	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	0	0	0	0	0
x_2	0	1.6	0	0	0	0
x_3	0	0	1.6	0	0	0
x_4	0	0	0	1.7	0	0
x_5	0	0	0	0	1.7	0
x_6	0	0	0	0	0	1.5

- Important application:
  - Normalize similarity matrix

GRAPH NOTATION: Size

$G=(V,E)$:

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

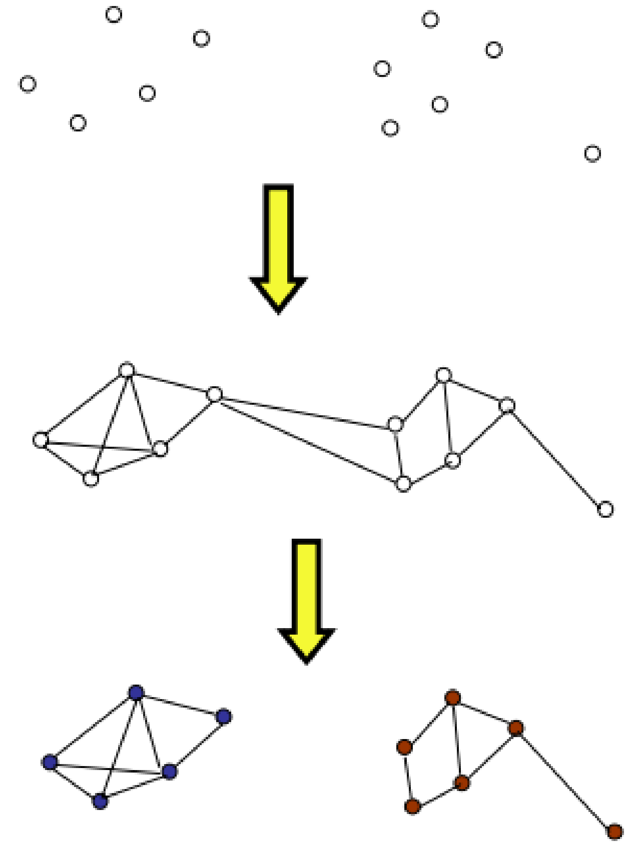


$|A|$:= the number of vertices in A

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

- 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 $A_1 \cup \cdots \cup A_k = V$.

GRAPH NOTATION: Graph Cut



First - graph representation of data

(largely, application dependent)

Then - graph partitioning

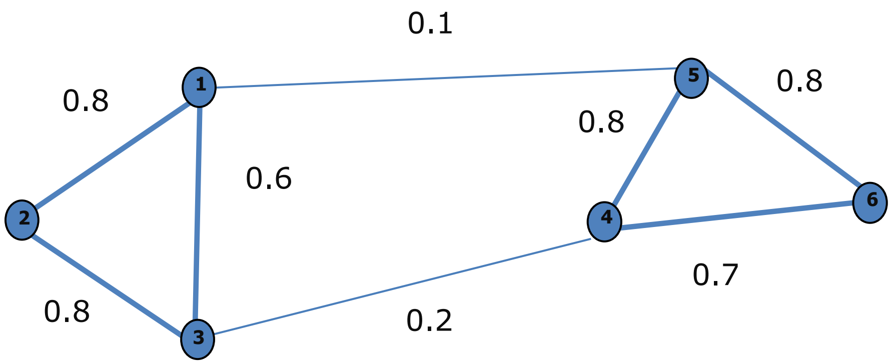
Disconnected graph components ⟷ Groups of points(Weakly connections in between components Strongly connections within components)

- Brief Clustering Review
- Graph Terminology
- **Graph Laplacian**
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Practical Details

GRAPH LAPLACIANS

- Un-normalized Graph Laplacian

$$L = D - W$$



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	0	0	-0.2	1.7	-0.8	-0.7
x_5	-0.1	0	0	-0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

- Important properties:
  - Eigenvalues are non-negative real numbers
  - Eigenvectors are real and orthogonal
  - Eigenvalues and eigenvectors provide an insight into the connectivity of the graph

GRAPH LAPLACIANS

- Un-normalized 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

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

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

GRAPH LAPLACIANS

- Un-normalized 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 $0 = \lambda_i \leq \lambda_2 \leq \cdots \leq \lambda_n$.

GRAPH LAPLACIANS

- Un-normalized 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 (number) 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.

GRAPH LAPLACIANS

- Normalized Graph Laplacian

$$L_{sym} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

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

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.

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut Point of View
- Practical Details

ALGORIGHM

Main trick is to change the representation of the abstract data points x_i to points $y_i \in \Re^k$: unfold the manifold

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

ALGORIGHM

- 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 $u_1, \ldots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

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

ALGORIGHM

- 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 $u_1, \ldots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ 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 $Lu = \lambda Du$.*

ALGORIGHM

- Normalized Graph Laplacian $L_{sym} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$

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 $u_1, \ldots, u_k$ of L_{sym}.
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ 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 $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

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

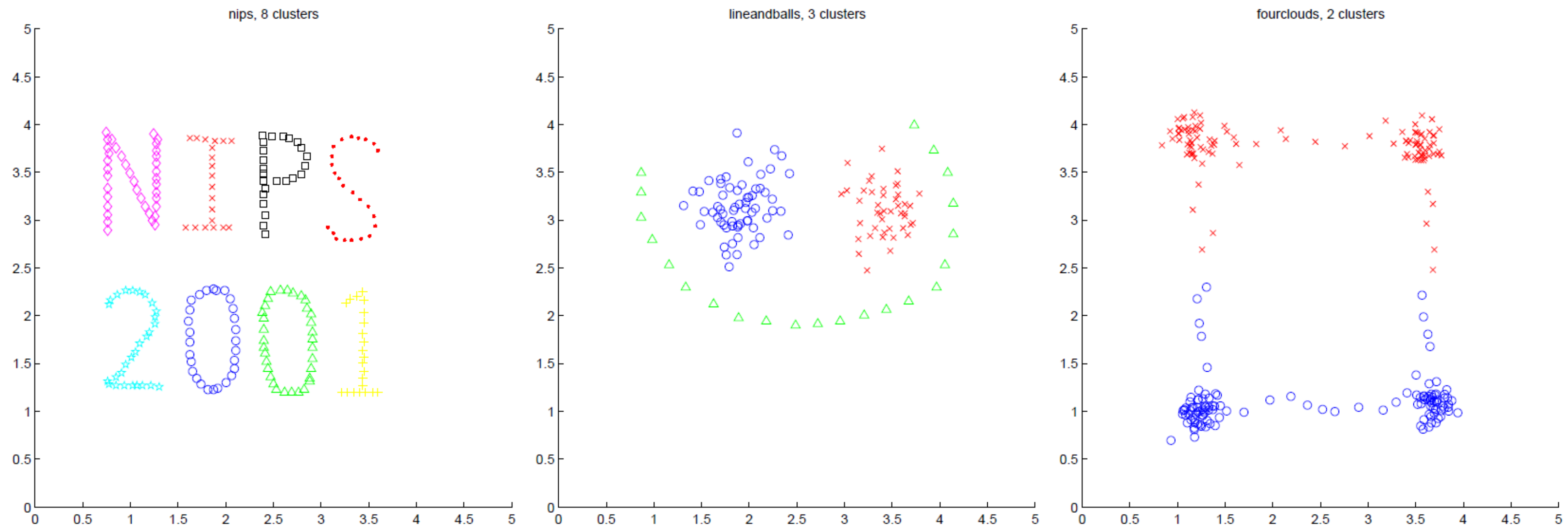
ALGORIGHM

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

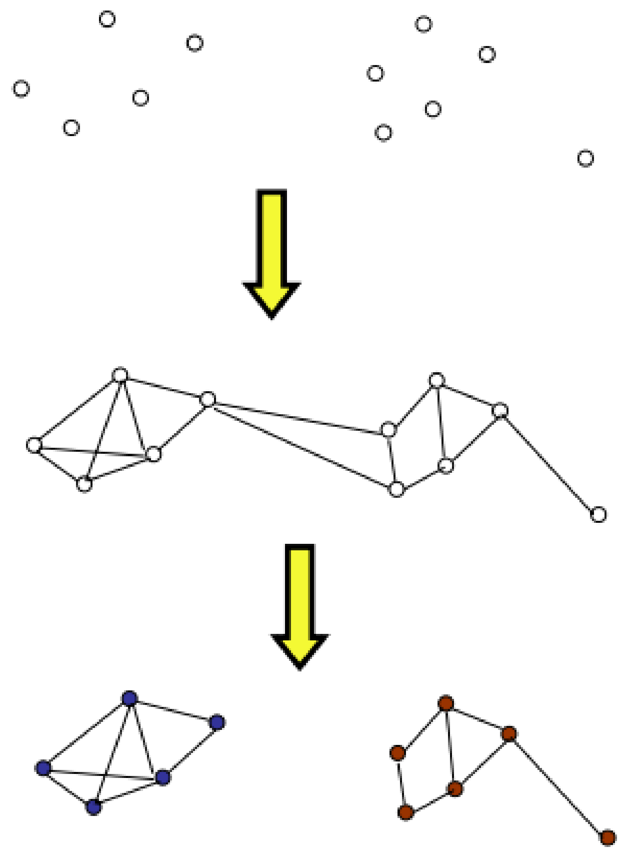


(a) (b) (c)

Spectral Clustering

- Brief Clustering Review
- Similarity Graph
- Graph Laplacian
- Spectral Clustering Algorithm
- Graph Cut 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

$G=(V,E)$:

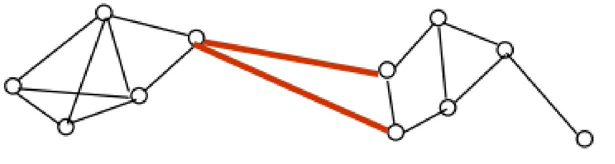
- For two not necessarily disjoint set $A, B \subset V$, we define

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

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

$$cut(A_1, \ldots, A_k) \coloneqq \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i})$$

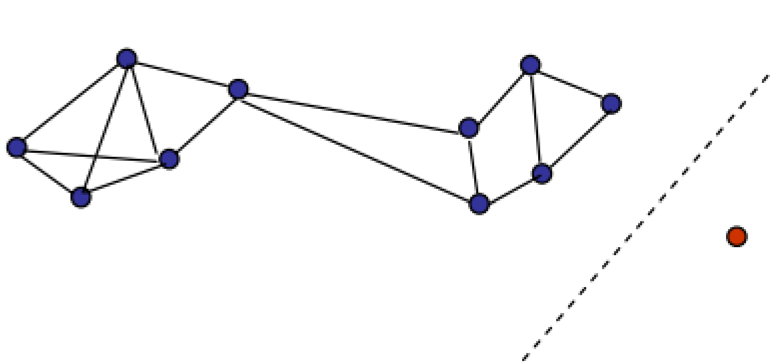
Cut between 2 sets $cut(A_1, A_2) = \sum_{n \in A_1} \sum_{m \in A_2} w_{nm}$



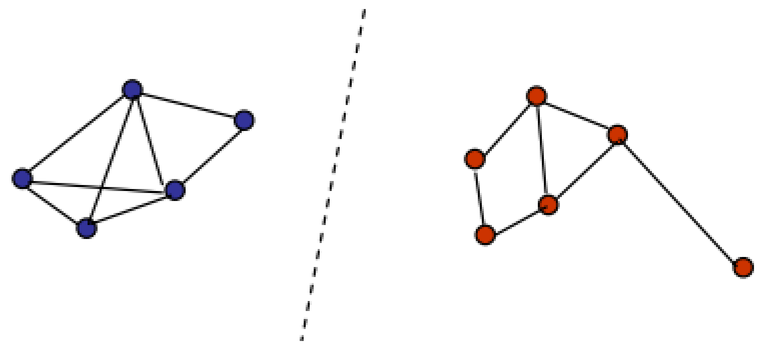
GRAPH CUT

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)

$$RatioCut(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

- Ncut(Shi and Malik, 2000): normalized cut
- Ncut for image segmentation: 20 best papers in 21 centuray in CV

$$Ncut(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{vol(A_i)} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{vol(A_i)}$$

GRAPH CUT

Problem!!!

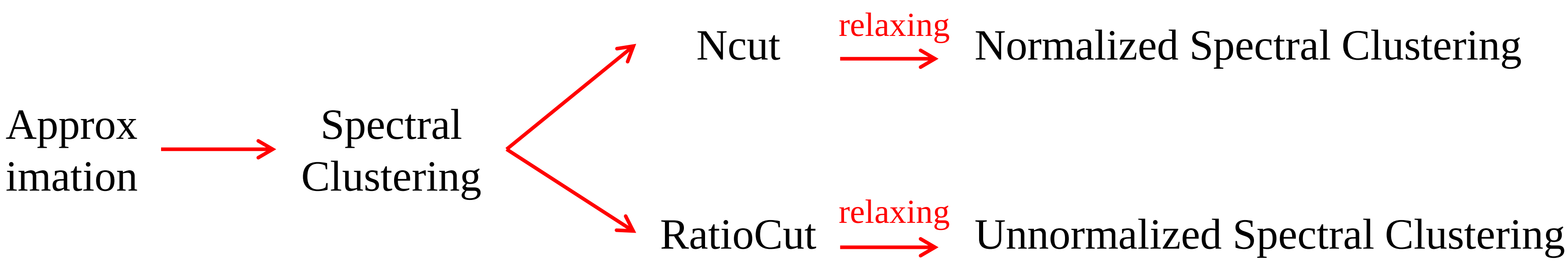
$$RatioCut(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

$$Ncut(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{vol(A_i)} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{vol(A_i)}$$

- NP hard

Solution!!!

- Approximation



GRAPH CUT

- Approximation RatioCut for k=2

Our goal is to solve the optimization problem:

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

Rewrite the problem in a more convenient form:

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

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

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

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

$$\begin{aligned} f'Lf &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\ &= \frac{1}{2}\sum_{i \in A, j \in \overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 + \frac{1}{2}\sum_{i \in \overline{A}, j \in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\ &= \mathrm{cut}(A, \overline{A})\left(\frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} + 2\right) \\ &= \mathrm{cut}(A, \overline{A})\left(\frac{|A| + |\overline{A}|}{|A|} + \frac{|A| + |\overline{A}|}{|\overline{A}|}\right) \\ &= |V| \cdot \mathrm{RatioCut}(A, \overline{A}). \end{aligned}$$

GRAPH CUT

- Approximation RatioCut for k=2

Additionally, we have

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

i.e.: f^T 𝟙 =0. The vector *f* as defined before is orthogonal to the constant one vector 𝟙 .

f satisfies

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

GRAPH CUT

- Approximation RatioCut for k=2

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

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

$$f'Lf = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$
$$= |V| \cdot \text{RatioCut}(A, \overline{A}).$$

$$\min_{A \subset V} f'Lf \text{ subject to } f \perp \mathbb{1} \quad \|f\| = \sqrt{n}.$$

Relaxation !!!

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

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

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

Only works for k = 2

f_i **as points in** R **and do K-means**

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

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

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

$$h_i' L h_i = \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

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

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

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

GRAPH CUT

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

- Approximation RatioCut for arbitrary k

Problem reformulation:

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

⇕

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

⇓ **Relaxation !!!**

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

⇓ **Rayleigh-Ritz Theorem**

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

GRAPH CUT

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

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

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

Similar to above one can check that:

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

GRAPH CUT

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

- Approximation Ncut for k=2

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

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

$$\min_{A} f'Lf \quad subject\ to\ f\ as\ in\ (6), Df \perp \mathbb{1}, f'Df = vol(V)$$

Relaxation !!!

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

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

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

Rayleigh-Ritz Theorem!!!

GRAPH CUT

- Approximation Ncut for arbitrary k

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

Problem reformulation:

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

$$\Updownarrow$$

$$\min_{A_1,\ldots,A_k} Tr(H'LH) \ \ subject\ to\ H'DH = I$$

$$\Downarrow$$

Relaxation !!!

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

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

$$\Downarrow$$

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
- Practical Details

PRACTICAL DETAILS

- Which graph Laplacian should be used?

  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 obejctive.

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.

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 Clustring. http://videolectures.net/mlcued08_azran_mcl/

SPECTRAL CLUSTERING

Thank you