Stat 321: Transposable Data
Spectral clustering

Art B. Owen

Stanford Statistics

Spectral clustering

- Relatively new method.
- Lots of promise.
- Defined via NP-hard graph criteria.

Spectral clustering

- Relatively new method.
- Lots of promise.
- Defined via NP-hard graph criteria.
- Approximate solution via matrix theory and eigenvectors.

Spectral clustering

- Relatively new method.
- Lots of promise.
- Defined via NP-hard graph criteria.
- Approximate solution via matrix theory and eigenvectors.
- Applications to information retrieval and image segmentation and network analysis.
- Several competing flavors.

Spectral clustering

- Relatively new method.
- Lots of promise.
- Defined via NP-hard graph criteria.
- Approximate solution via matrix theory and eigenvectors.
- Applications to information retrieval and image segmentation and network analysis.
- Several competing flavors.
- Good news = bad news = we still have to think about our data

Spectral clustering

Main reference

I mostly follow the excellent exposition of Ulrike von Luxborg

Spectral clustering

Main reference

I mostly follow the excellent exposition of Ulrike von Luxborg

Graphs

- $G = (V, E)$
- Vertices v_i $i = 1, \dots, n$. $v_i \in V$
- Edges are vertex pairs from $V \times V$
- Undirected and weighted
- Represent by $w_{ij} = w_{ji} \geq 0$.
- $w_{ij} > 0$ iff G has an ij edge

Spectral clustering

Main reference

I mostly follow the excellent exposition of Ulrike von Luxborg

Graphs

- $G = (V, E)$
- Vertices v_i $i = 1, \ldots, n$. $v_i \in V$
- Edges are vertex pairs from $V \times V$
- Undirected and weighted
- Represent by $w_{ij} = w_{ji} \geq 0$.
- $w_{ij} > 0$ iff G has an ij edge

Graph clustering

- Partition the vertices
- With large weights within and small weights between

Graph Cut

Binary split

- $A \subset V$ and $A^c = V - A$
- $$\mathsf{Cut}(A, A^c) = \sum_{i \in A} \sum_{j \in A^c} w_{ij}$$
- Pick A to minimize Cut

Graph Cut

Binary split

- $A \subset V$ and $A^c = V - A$
- $$\text{Cut}(A, A^c) = \sum_{i \in A} \sum_{j \in A^c} w_{ij}$$
- Pick A to minimize Cut, often get singleton A

Graph Cut

Binary split

- $A \subset V$ and $A^c = V - A$
- $$\mathsf{Cut}(A, A^c) = \sum_{i \in A} \sum_{j \in A^c} w_{ij}$$
- Pick A to minimize Cut, often get singleton A
- Penalize small groups via group size $|A|$ to favor balance

$$\mathsf{RatioCut}(A, A^c) = \sum_{i \in A} \sum_{j \in A^c} w_{ij} \Bigl(\frac{1}{|A|} + \frac{1}{|A^c|} \Bigr)$$

- Best split hard to find

Lets relax

Define $f \in \mathbb{R}^n$

$$f_i = \begin{cases} \sqrt{|A^c|/|A|}, & i \in A \\ -\sqrt{|A|/|A^c|}, & i \in A^c \end{cases}$$

NB: $\sum_i f_i = 0,$ and $\sum_i f_i^2 = |V|$

Lets relax

Define $f \in \mathbb{R}^n$

$$f_i = \begin{cases} \sqrt{|A^c|/|A|}, & i \in A \\ -\sqrt{|A|/|A^c|}, & i \in A^c \end{cases}$$

NB: $\sum_i f_i = 0$, and $\sum_i f_i^2 = |V|$

Now

$$\begin{aligned} \sum_{ij} w_{ij}(f_i - f_j)^2 &= \sum_{i\in A}\sum_{j\in A^c}(w_{ij}+w_{ji})\left(\sqrt{\frac{|A|}{|A^c|}}+\sqrt{\frac{|A^c|}{|A|}}\right)^2 \\ &= 2\text{Cut}(A, A^c)\left(\frac{|A^c|}{|A|}+\frac{|A|}{|A^c|}+2\right) \\ &= 2\text{Cut}(A, A^c)\left(\frac{|A^c|+|A|}{|A|}+\frac{|A|+|A^c|}{|A^c|}\right) \\ &= 2|V|\text{RatioCut}(A, A^c) \end{aligned}$$

Relaxed problem

Minimize

$\sum_{ij} w_{ij}(f_i - f_j)^2$ subject to

1. $\sum_i f_i = 0$
2. $\sum_i f_i^2 = |V|$

But forgetting about the combinatorial constraint

Relaxed problem

Minimize

$\sum_{ij} w_{ij}(f_i - f_j)^2$ subject to

1. $\sum_i f_i = 0$
2. $\sum_i f_i^2 = |V|$

But forgetting about the combinatorial constraint

Solution

Via an eigen vector algorithm. The smallest eigen value is 0 f_i is the eigen vector for the second smallest eigen value

Then take $A = \{i \mid f_i \geq 0\}$

Relaxed problem

Minimize

$\sum_{ij} w_{ij}(f_i - f_j)^2$ subject to

1. $\sum_i f_i = 0$
2. $\sum_i f_i^2 = |V|$

But forgetting about the combinatorial constraint

Solution

Via an eigen vector algorithm. The smallest eigen value is 0 f_i is the eigen vector for the second smallest eigen value

Then take $A = \{i \mid f_i \geq 0\}$

Variants

- How to pick w_{ij}
- Alternatives to RatioCut
- Binary splits other than the sign and k fold splits

Size of sets

$d_i = \sum_j w_{ij}$ generalizes degree of i
For $A \subseteq V$

- $|A|$ = cardinality of A
- $\mathrm{vol}(A) = \sum_{i \in A} d_i$

Size of sets

$d_i = \sum_j w_{ij}$ generalizes degree of i
For $A \subseteq V$

- $|A|$ = cardinality of A
- $\mathrm{vol}(A) = \sum_{i \in A} d_i$

From points to vertices

We will represent points x_i as vertices v_i
$\|x_i - x_j\|$ small will imply w_{ij} large.
Splitting the graph clusters the points.

Size of sets

$d_i = \sum_j w_{ij}$ generalizes degree of i
For $A \subseteq V$

- $|A| =$ cardinality of A
- $\mathrm{vol}(A) = \sum_{i \in A} d_i$

From points to vertices

We will represent points x_i as vertices v_i
$\|x_i - x_j\|$ small will imply w_{ij} large.
Splitting the graph clusters the points.

Similarity measures for $v_i \equiv x_i \in \mathbb{R}^d$

- ϵ neighborhood $w_{ij} = 1_{\|x_i - x_j\| \le \epsilon}$
- k-NN graph $w_{ij} = 1$ if i is one of j's k NNs (or conversely)
- $w_{ij} = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$

Graph Laplacian(s)

Graph Laplacian matrix (unweighted)

$$L = D - W$$

$$D = \text{diag}(d_1, \dots, d_n) \quad \text{degree matrix}$$

Graph Laplacian(s)

Graph Laplacian matrix (unweighted)

$$L = D - W$$

$$D = \mathrm{diag}(d_1, \dots, d_n) \quad \text{degree matrix}$$

Properties

L is symmetric and positive semidefinite

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

Smallest eigenvalue is 0, corresponding eigenvector is $(1, \dots, 1) \in \mathbb{R}^n$

Graph Laplacian(s)

Graph Laplacian matrix (unweighted)

$$L = D - W$$

$$D = \text{diag}(d_1, \dots, d_n) \quad \text{degree matrix}$$

Properties

L is symmetric and positive semidefinite

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

Smallest eigenvalue is 0, corresponding eigenvector is $(1, \dots, 1) \in \mathbb{R}^n$

We're interested in smallest eigenvalues of L (largest of $W - D$)
$0 \le \lambda_1 \le \cdots \le \lambda_n$

Graph Laplacian

Components

G has k connected components $\implies$ L has k eigenvalues of 0
Sort edges into groups, then

$$L = \text{diag}(L_1 \quad L_2 \quad \dots \quad L_k)$$

Each L_j has an eigen value of 0

Graph Laplacian

Components

G has k connected components $\implies$ L has k eigenvalues of 0

Sort edges into groups, then

$$L = \text{diag}(L_1 \quad L_2 \quad \dots \quad L_k)$$

Each L_j has an eigen value of 0

Normalizations

Symmetric normalization

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

Random walk normalization

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

L_{ij} gives probability of graph walking to j from i

Properties of L_{sym} and L_{rw}

von Luxborg

- $f' L_{\text{sym}} f = \frac{1}{2} \sum_{ij} w_{ij} \Bigl(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \Bigr)^2$
- $L_{\text{rw}} v = \lambda v \iff L_{\text{sym}} w = \lambda w$, for $w = D^{1/2} v$
- L_{rw} has eigval 0 for eig vec of 1s
- Both pos semidef
- $\#$ 0 eigvals is $\#$ connected components

Spectral clustering

Unnormalized

- Construct similarity graph W
- Get $L = D - W$
- Find smallest k eigenvalue/vector pairs
- Let V be the $n \times k$ eigvector matrix
- Represent point i by y_i i'th row of V
- Run k means on the y_i

Spectral clustering

Normalized (per Shi and Malik (2000))

- Construct similarity graph W
- Get $L = D - W$
- Find smallest k eigenvalue/vector pairs in generalized eigenvalue problem $Lv = \lambda Dv$
- Or $\cdots$ just use $L_{\mathrm{rw}} v = \lambda v$
- Let V be the $n \times k$ eigvector matrix
- Represent point i by y_i i'th row of V
- Run k means on the y_i

Spectral clustering

Normalized (per Ng, Jordan and Weiss (2002))

- Construct similarity graph W
- Get $L_{\mathsf{sym}} = I - D^{-1/2} W D^{-1/2}$
- Find smallest k eigenvalue/vector pairs of L
- Let V be the $n \times k$ eigvector matrix
- Get U by normalizing rows of V to unit length
- Represent point i by y_i i'th row of U
- Run k means on the y_i

Actually they run a clever k means that expects the cluster means to be mutually orthogonal
The extra normalization step helps when cluster sizes are very unequal.

k-group graph cuts

Seeking 'light' edges between 'heavy' edges within

$$\mathrm{Cut}(A_1,\dots,A_k) = \sum_{i=1}^{k} \mathrm{Cut}(A_i, A_i^c)$$

$$\mathrm{RatioCut}(A_1,\dots,A_k) = \sum_{i=1}^{k} \mathrm{Cut}(A_i, A_i^c)\frac{1}{|A_i|} \quad \text{Hagen Kahng 1992}$$

$$\mathrm{NCut}(A_1,\dots,A_k) = \sum_{i=1}^{k} \mathrm{Cut}(A_i, A_i^c)\frac{1}{\mathrm{vol}A_i} \quad \text{Shi Malik 2000}$$

We relaxed RatioCut to get unnormalized spectral clustering
Relaxing NCut gets normalized spectral clustering (Shi Malik version)

More

Guattery and Miller: cockroach graphs lead spectral clustering astray

More

Guattery and Miller: cockroach graphs lead spectral clustering astray

Random walks

Ncut$(A, A^c) = \Pr(A^c \mid A) + \Pr(A \mid A^c)$. Expected traffic between groups. 1st eigenvector describes stationary distribution. 2nd eigenvector describes correction: extra probability for $i \to j$ transitions after (large) m steps governed by $z_2 z_2'$. Going $i \to j$ slightly more likely if sign(z_{2i}) = sign(z_{2j}).

More

Guattery and Miller: cockroach graphs lead spectral clustering astray

Random walks

Ncut$(A, A^c) = \Pr(A^c \mid A) + \Pr(A \mid A^c)$. Expected traffic between groups. 1st eigenvector describes stationary distribution. 2nd eigenvector describes correction: extra probability for $i \to j$ transitions after (large) m steps governed by $z_2 z_2'$. Going $i \to j$ slightly more likely if sign(z_{2i}) = sign(z_{2j}).

Commute distance

Expected time to go from i to j and back

Almost but not quite the dist in spectral clustering

More

Guattery and Miller: cockroach graphs lead spectral clustering astray

Random walks

Ncut$(A, A^c) = \Pr(A^c \mid A) + \Pr(A \mid A^c)$. Expected traffic between groups. 1st eigenvector describes stationary distribution. 2nd eigenvector describes correction: extra probability for $i \to j$ transitions after (large) m steps governed by $z_2 z_2'$. Going $i \to j$ slightly more likely if sign(z_{2i}) = sign(z_{2j}).

Commute distance

Expected time to go from i to j and back
Almost but not quite the dist in spectral clustering

Perturbation theory

- Stable eigenvectors ...

More

Guattery and Miller: cockroach graphs lead spectral clustering astray

Random walks

Ncut$(A, A^c) = \Pr(A^c \mid A) + \Pr(A \mid A^c)$. Expected traffic between groups. 1st eigenvector describes stationary distribution. 2nd eigenvector describes correction: extra probability for $i \to j$ transitions after (large) m steps governed by $z_2 z_2'$. Going $i \to j$ slightly more likely if sign(z_{2i}) = sign(z_{2j}).

Commute distance

Expected time to go from i to j and back
Almost but not quite the dist in spectral clustering

Perturbation theory

- Stable eigenvectors . . .
- Come from well separated eigenvalues

Where to cut

k means using r eigenvectors

- k means with $r = k$
- k means with $r = k - 1$ (eg $k = 2$ only needs $r = 1$ eigenvector)
- If r eigenvectors $\rightarrow k = 2^r$ clusters ... take $r = \lceil \log_2(k) \rceil$

Where to cut

k means using r eigenvectors

- k means with $r = k$
- k means with $r = k - 1$ (eg $k = 2$ only needs $r = 1$ eigenvector)
- If r eigenvectors $\to k = 2^r$ clusters . . . take $r = \lceil \log_2(k) \rceil$

Other

- For $k = 2$, we can use direct cut-style measures instead of k-means
- Recursive bisection with or without k-means

Alternatives

Alternative dist

$$W_{ij} = \exp(-\beta\|x_i - x_j\|)$$

Instead of $\exp(-\beta\|x_i - x_j\|^2)$.
Gets 'path weight' $x_1 \to x_2 \to \cdots \to x_n$ of $\exp(-\beta\sum_i \|x_{i+1} - x_i\|)$.

Kannan Vempala Vetta

Use Cheeger conductance

$$\phi(A, A^c) = \frac{\mathsf{Cut}(A, A^c)}{\min(\mathsf{vol}(A), \mathsf{vol}(A^c))}$$

Directed graphs

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

is symmetric in W. So are size penalties based on $\mathsf{Cut}(A, A)$.

Clustering examples

Examples

Show figure from Ng, Jordan and Weiss

Notes

- Spectral clustering soundly beats k-means on straggly arbitrary shaped clusters
- It even beats single linkage in such examples
- The reason is that having 5 connections at distance $d + \epsilon$ counts for more than having just one at d
- We might expect 'reverse counter-examples' for the other methods.

References

- Ulrike von Luxborg: Excellent and very clear tutorial on spectral clustering
- Chris Ding: Two well illustrated ICML tutorials online
- Ng, Jordan, Weiss: Concise and well illustrated NIPS paper
- Shortreed and Meila: Graph examples with random walk interpretations