Certifying Global Optimality of Graph Cuts via Semidefinite Relaxation

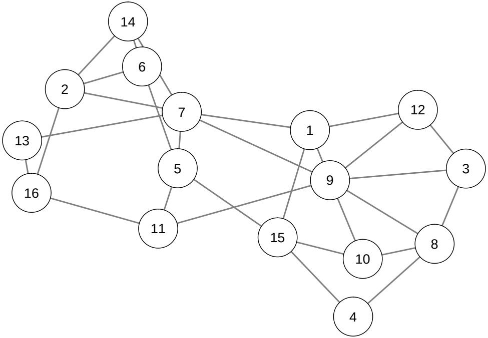
A Performance Guarantee for Spectral Clustering

Shuyang Ling and Thomas Strohmer

Feb 18, 2020

Introduction

Graph $G = (V, E)$ consists of a set of vertices V and a set of edges E connecting vertices

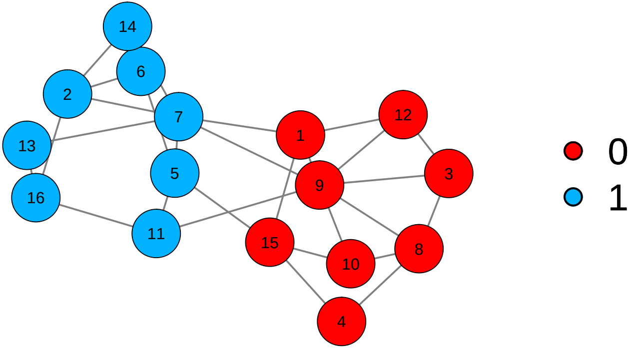


Summarized by edge weight matrix W

w_{ij} is the value of the edge between vertices v_i and v_j

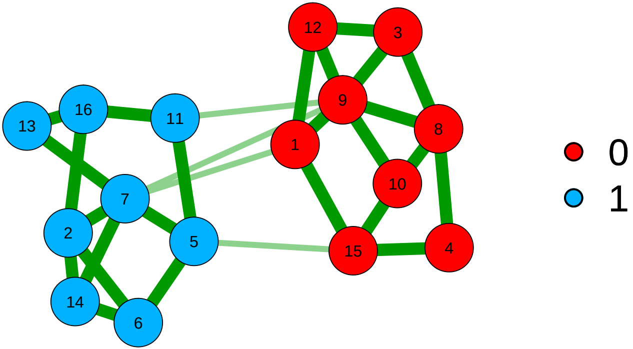
Graph Partitioning

What is the “best” way to cluster the vertices?



Graph Cut Objectives

Find a “reasonable” way to cut edges to obtain two disconnected subgraphs



Graph Cut Objectives

Let

- $\{C_1, ..., C_k\}$ be a partition of V
- $cut(C, C^c) = \sum_{v_i \in C} \sum_{v_j \in C^C} w_{ij}$
- $|C|$ = number of vertices in set C
- $vol(C)$ = sum of edges pointing to vertices in set C

Plausible graph cut objectives

- Min Cut: $\min_j \sum_j^k cut(C_j, C_j^c)$
- Ratio Cut: $\min_j \sum_j^k \frac{cut(C_j, C_j^c)}{|C_j|}$
- Normalized Cut: $\min_j \sum_j^k \frac{cut(C_j, C_j^c)}{vol(C_j)}$

Graph Cut Objectives

Let

- $\{C_1, ..., C_k\}$ be a partition of V
- $cut(C, C^c) = \sum_{v_i \in C} \sum_{v_j \in C^C} w_{ij}$
- $|C|$ = number of vertices in set C
- $vol(C)$ = sum of edges pointing to vertices in set C

Plausible graph cut objectives

- Min Cut: $\min_j \sum_j^k cut(C_j, C_j^c)$
- **Ratio Cut**: $\min_j \sum_j^k \frac{cut(C_j, C_j^c)}{|C_j|}$
- Normalized Cut: $\min_j \sum_j^k \frac{cut(C_j, C_j^c)}{vol(C_j)}$

Ratio Cut Reformulation

Define

- Degree matrix D s.t. $d_{ii} = \sum_j w_{ij}$
- Combinatorial graph Laplacian $L = D - W$
- Cluster membership matrix $U \in \mathbb{R}^{n \times k}$ s.t.
  $$u_{ij} = \begin{cases} n_j^{-1/2} & v_i \in C_j \\ 0 & \text{else} \end{cases}$$

Ratio cut objective:

$$\begin{aligned} \arg\min_U \ & \mathsf{Tr}(U^\top L U) \\ \text{s.t. } & u_{ij} = \begin{cases} n_j^{-1/2} & v_i \in C_j \\ 0 & \text{else} \end{cases} \end{aligned}$$

Ratio Cut Reformulation

Ratio cut objective: $\arg\min_U \ \mathrm{Tr}(U^\top L U)$

- U is restricted to a very specific form
- $U^\top U = I_k$
- $UU^\top = Z \in \mathbb{R}^{n\times n}$ s.t. $z_{ij} = \begin{cases} n_k^{-1} & v_i, v_j \text{ in same cluster } C_k \\ 0 & \text{else} \end{cases}$
- $\mathrm{Tr}(U^\top U) = \mathrm{Tr}(UU^\top) = k$

Spectral Clustering

- Ratio cut problem is a discrete optimization problem that is NP-hard
- Remove the discreteness constraint

$$\arg\min_U \operatorname{Tr}(U^\top L U)$$
$$\text{s.t. } U^\top U = I_k$$

- Rayleigh-Ritz theorem: $\hat{U} = \begin{bmatrix} u_0 & \cdots & u_{k-1} \end{bmatrix}$ first k eigenvectors of L corresponding to the k smallest eigenvalues
- $\hat{U}$ doesn't provide cluster assignments
- Treat $\hat{U}$ as an embedding in $\mathbb{R}^{k-1}$ and perform k-means clustering
- Not guaranteed to correspond to the ratio cut solution
- Not much theoretical basis for why/how this works

Spectral Clustering: Justification 1

Davis-Kahan theorem: If two matrices A and $\tilde{A}$ are "close", then the subspaces generated by their eigenvectors is also "close"

Let

- $G_{iso} = (V, E_{iso})$ be a graph with k disconnected subgraphs
- W_{iso} be the corresponding weight matrix
- $L_{iso} = D_{iso} - W_{iso}$ be the corresponding Laplacian matrix

Then

- L_{iso} has k zero eigenvalues
- The k eigenvectors of L_{iso} that correspond to the k zero eigenvalues are of the form $u_{ij} = \begin{cases} n_j^{-1/2} & v_i \in \text{ subgraph } j \\ 0 & \text{else} \end{cases}$

This is the type of U we want to find in the ratio cut problem

Spectral Clustering: Justification 1

Davis-Kahan theorem: If two matrices A and $\tilde{A}$ are "close", then the subspaces generated by their eigenvectors is also "close"

Let

- $W = W_{iso} + \Delta$ such that Δ is "small" and W corresponds to a connected graph $G = (V, E)$
- $L = D - W$ be the combinatorial Laplacian matrix

Then

- L has one zero eigenvector
- The eigenspaces generated by the eigenvectors of L and L_{iso} is "small"
  - Hopefully the eigenvectors of L are a good approximation of the eigenvectors of L_{iso}

Spectral Clustering: Justification 2

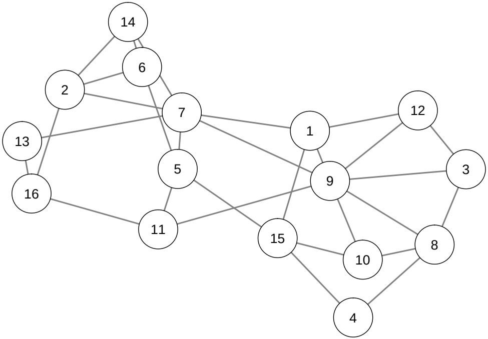
Laplacian eigenmap - the embedding generated by the eigenvalues and eigenvectors of L

- If instead of $\hat{U} = \begin{bmatrix} u_0 & \cdots & u_{k-1} \end{bmatrix}$, we use $\hat{U} = \begin{bmatrix} u_1/\sqrt{\lambda_1} & \cdots & u_{k-1}/\sqrt{\lambda_{k-1}} \end{bmatrix}$, we get a Laplacian eigenmap
- Characterizes the expected commute time of a random walk between two vertices
- Equivalent to performing kernel k-means on the generalized inverse of L:

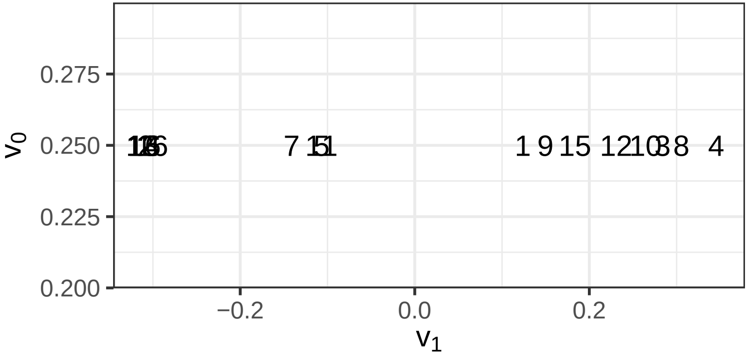
$$\arg\max_{U} \mathsf{Tr}(U^\top L^\dagger U)$$

$$\text{s.t. } u_{ij} = \begin{cases} n_j^{-1/2} & v_i \in C_j \\ 0 & \text{else} \end{cases}$$

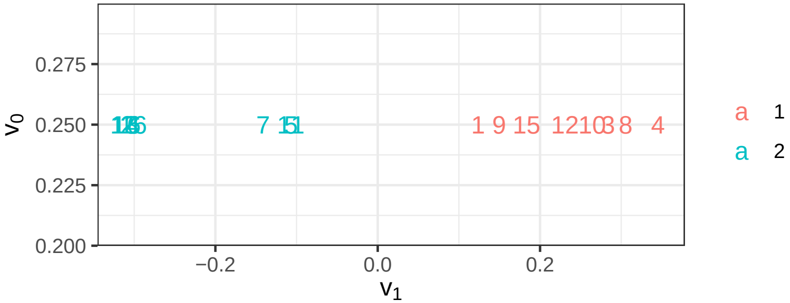
Example



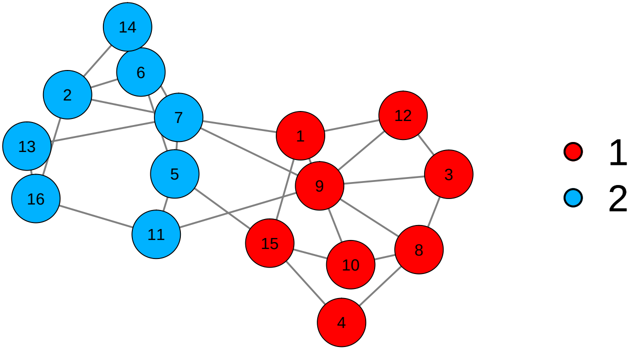
Example



Example



Example



RatioCut-SDP

- Main idea: Spectral clustering is too loose of a relaxation of the ratio cut problem
- Solution: Add additional constraints
- Goal: Find out when the solution to the relaxed version of the ratio cut problem coincides with the actual solution

RatioCut-SDP

Recall the ratio cut problem: $\arg\min_U \; \mathrm{Tr}(U^\top L U)$

Rewrite $\mathrm{Tr}(U^\top L U) = \mathrm{Tr}(L U U^\top) = \mathrm{Tr}(LZ)$

Then we get

$$\begin{aligned} &\arg\min_Z \; \mathrm{Tr}(LZ) \\ &\text{s.t. } z_{ij} = \begin{cases} n_k^{-1} & v_i, v_j \in C_k \\ 0 & \text{else} \end{cases} \end{aligned}$$

- Still a discrete optimization problem
- Also note that $Ze = e$, Z is positive semidefinite, $\mathrm{Tr}(Z) = k$, entries of $Z \geq 0$

RatioCut-SDP

$$
\begin{aligned}
\arg\min_{Z}\ & \mathsf{Tr}(LZ) \\
\text{s.t. } & Z \geq 0 \text{ element-wise} \\
& Z \text{ is PSD} \\
& Ze = e \\
& \mathsf{Tr}(Z) = k
\end{aligned}
$$

This is a convex semidefinite programming problem

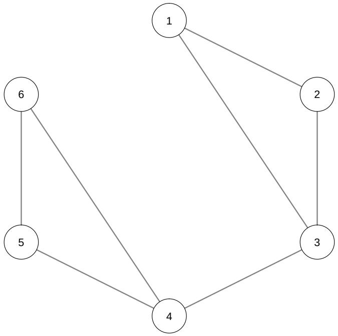
RatioCut-SDP

- Spectral clustering has no (known) conditions for which we are guaranteed the correct clustering

Theorem 3.1

- Given
  - Connected graph $G = (V, E)$ and corresponding weight matrix W
  - Number of clusters k
  - Known optimal ratio cut clustering
  - W_{iso} based on W and the known optimal ratio cut clustering of G
  - D and L corresponding to W
  - D_{iso} and L_{iso} corresponding to W_{iso}
  - $D_\delta = D - D_{iso}$
- Then
  - $\{C_1, ..., C_k\} = RCSDP(L, k)$ is the optimal ratio cut clustering given that $||D_\delta||_{op} < \frac{1}{4}\lambda_{k+1}(L_{iso})$

Example



Example

```r
L <- graph.laplacian(W)
rc.sdp(L, 2)$Z %>%
  MASS::fractions()
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1/3  1/3  1/3     0    0    0
[2,] 1/3  1/3  1/3     0    0    0
[3,] 1/3  1/3  1/3     0    0    0
[4,]    0    0    0 1/3  1/3  1/3
[5,]    0    0    0 1/3  1/3  1/3
[6,]    0    0    0 1/3  1/3  1/3
```

Example

Check conditions

```r
W.iso <- W
W.iso[3, 4] <- W.iso[4, 3] <- 0
L.iso <- graph.laplacian(W.iso)
D <- diag(colSums(W))
D.iso <- diag(colSums(W.iso))
D.delta <- D - D.iso
n <- nrow(W)

norm(D.delta, type = '2')
```

```
[1] 1
```

```r
eigen(L.iso)$values[n - 2] / 4
```

```
[1] 0.75
```

Example

- In general, the output of RatioCut-SDP $\hat{Z}$ does not provide cluster memberships
- Not clear how to obtain clustering from $\hat{Z}$
- One idea: Use SVD/spectral decomposition on $\hat{Z}$ to obtain an embedding and perform k-means on the embedding
- Ideally we could obtain $U \in \mathbb{R}^{n \times k}$ from $\hat{Z}$ but in general $rank(Z) = n - 1$
- Rank constraints are not convex, no good way to force $Z = UU^\top$ for $U \in \mathbb{R}^{n \times k}$k