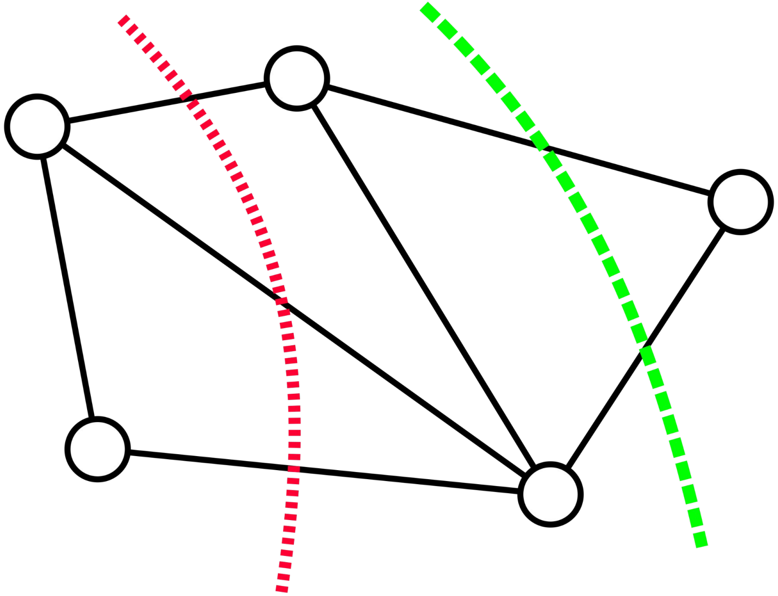
Community Detection Lecture 2

E0: 259

Cuts

- Red line - cut has 3 edges
- Green line - cut has 2 edges.
- Intuition - similar to edge betweenness
- Intuition from max flow, min cut theorem



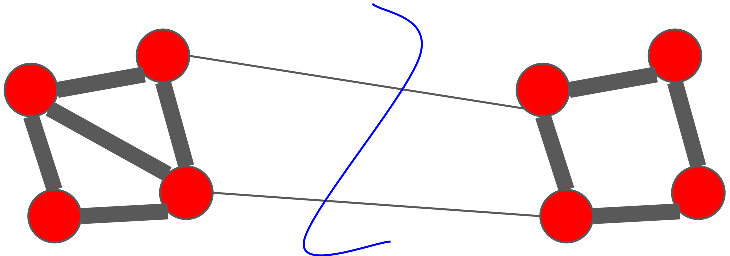
Cut based Partition Definitions

$$G = (V,E)\,,\ V = V_1 + V_2, \qquad cut(V_1, V_2) = \sum_{i \in V_1,\, j \in V_2} e_{ij}\,, \qquad Vol(V_1) = \sum_{i \in V_1} k_i$$

$$Ratio\ Cut: \qquad Q = \frac{cut(V_1, V_2)}{\|V_1\|} + \frac{cut(V_1, V_2)}{\|V_2\|}$$

$$Normalized\ cut: \qquad Q = \frac{cut(V_1, V_2)}{Vol(V_1)} + \frac{cut(V_1, V_2)}{Vol(V_2)}$$

$$Conductance: \qquad Q = \frac{cut(V_1, V_2)}{min(Vol(V_1), Vol(V_2))}$$

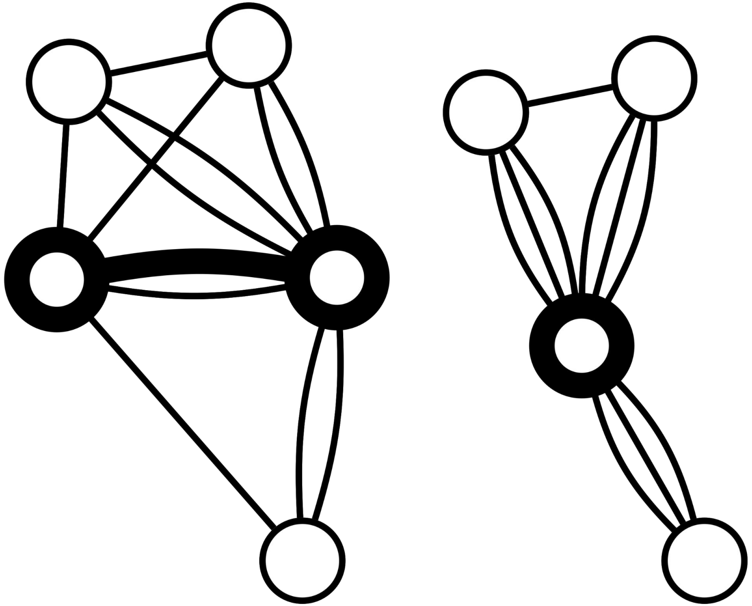


Graph Partitioning Problem

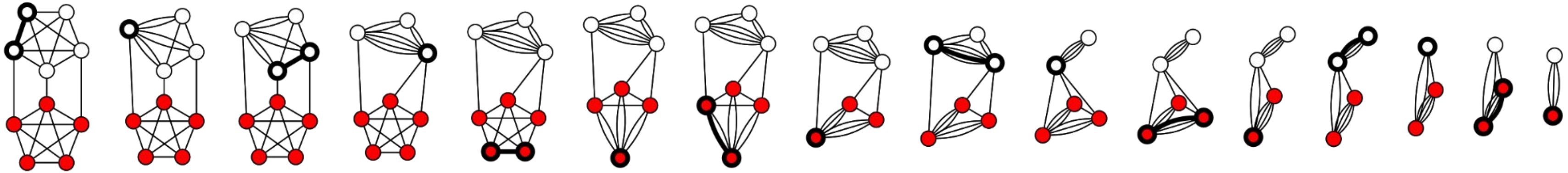
- NP hard
- Different approximation algorithms
- We will focus on randomized algorithms and spectral decomposition based algorithms

Randomized Algorithm - Karger’s Algorithm

- Pick an edge at random
- Merge the nodes, retain edges
- Repeat until only two nodes left.
- Edges between these two represent a cut.



One run of Karger’s algorithm



Properties of Karger’s Algorithm

$$P(cut = mincut) = \frac{2}{n^2}$$

Run the algorithm Ώ(n^2) times and pick minimum cut!

Algorithm 1 Karger's Min Cut Algorithm

Input: $G = (V, E)$
Output: set of edges which is min cut

$mincut = \phi$
$mincut_k = \infty$

for $i = 1 : \Omega(N^2)$
$cut_G = G$

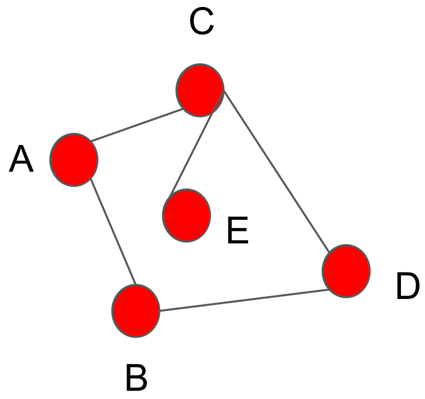
    while $|nodes(cut_G)| > 2$
        pick random $e \in E \in cut_G$,
        $cut_G = merge(cut_G, e)$

    if $|cut_G(E)| < mincut_k$
        $mincut_k = |edges(cut_G)|$
        $mincut = edges(cut_G)$

Spectral Decomposition Algorithms

- Any graph can be represented via an adjacency matrix
-

$$
\begin{array}{cc}
\begin{array}{ccccc} A & B & C & D & E \end{array} & \\
\begin{pmatrix}
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{pmatrix} & \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array}
\end{array}
$$



A Simple Idea and Some Linear Algebra

- Let $G = (V, E)$
- Consider a partition $V = V_1 + V_2$
- $\forall i \in V_1, s_i = +1$
  $\forall i \in V_2, s_i = -1$
- For any edge e_{ij}, what can we say about $(s_i - s_j)^2$?
- if i and j belong to same partition , $(s_i - s_j)^2 = 0$,
- else, $(s_i - s_J)^2 = 4$

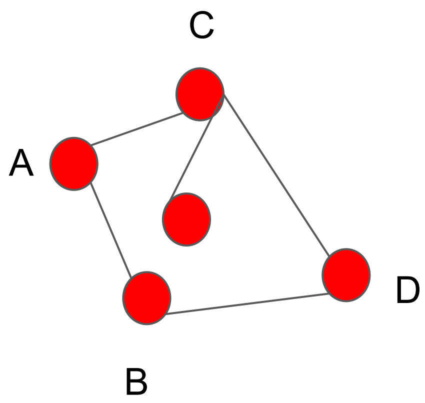
- So what is $cut(V_1, V_2)$, i.e., number of edges going from nodes in V_1 to V_2?
- $cut(V_1, V_2) = \frac{1}{4} \sum_{e_{ij}} (s_i - s_j)^2$
- Definition: $\delta_{ij} = 0 if i \neq j, 1$ otherwise
- Can we rewrite the size of the cut in terms of the adjacency matrix and the Kronecker delta function?
- Lets give it a shot

- $\frac{1}{4}\sum_{e_{ij}}(s_i - s_j)^2 = \frac{1}{8}\sum_{i,j} A_{ij}(s_i - s_j)^2$
- $= \frac{1}{8}\sum_{i,j} A_{ij}(s_i^2 + s_j^2 - 2s_is_j)$
- Therefore, $cut(V_1, V_2) = \frac{1}{4}\sum_{i,j}(k_i\delta_{ij}s_i^2 - A_{ij}s_is_j)$
- $= \frac{1}{4}\sum_{i,j}(k_i\delta_{ij} - A_{ij})s_is_j$
- $= \frac{1}{4}\sum_{i,j}(D_{ij} - A_{ij})s_is_j$

The Graph Laplacian

- $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- $\mathbf{A} =$

$$\begin{array}{cc} \begin{matrix} A & B & C & D & E \end{matrix} & \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} & \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \end{array}$$



- $\mathbf{L} =$

$$\begin{array}{cc} \begin{matrix} A & B & C & D & E \end{matrix} & \\ \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} & \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \end{array}$$

Balanced Min Cut with Graph Laplacian

- $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- $cut(V_1, V_2) = \frac{1}{4}\sum_{i,j}(D_{ij} - A_{ij})s_i s_j = \frac{\mathbf{s^T L s}}{4}$
- We want a minimum cut!
- $min\frac{\mathbf{s^T L s}}{4}$
- let us also seek equal number of nodes in each partition to keep it balanced
- contraint: $\sum_i s_i = 0, s_i = \pm 1$
- Integer minimization problem, can't guarantee quick computation, a.k.a NP hard!

Solve an Easier Problem that is a Good Approximation

- Take away constraint that $s_i = \pm 1$ and allow it to take any real number value.
- $min \frac{\mathbf{x}^{\mathbf{T}} \mathbf{L} \mathbf{x}}{4}$
- keep following constraints
- contraints: (i) $\sum_i x_i = 0, x_i \in R$, (ii) $\sum_i x_i^2 = n$
- If we can find x_i efficiently, then set $s_i = sign(x_i)$

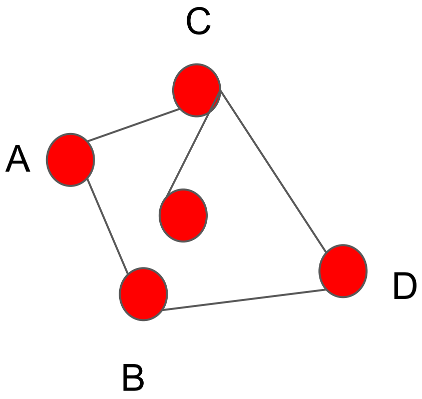
High School Math to the Rescue!

- Lagrange multipliers: $min \frac{\mathbf{x^T L x}}{4} - \lambda(\mathbf{x^T x} - \mathbf{n}), \mathbf{x^T e} = \mathbf{0}$
- How do you solve this? (i) Differentiate w.r.t $\mathbf{x}$, (ii) Differentiate w.r.t λ and set each to 0.
- (i) gives $\mathbf{Lx} = \lambda \mathbf{x}$ - the Eigenvalue problem!!
- what is the smallest eigenvalue and eigenvector?
- consider $\mathbf{x_i} = \mathbf{e}$
- what is $\mathbf{Le}$?

The Graph Laplacian

- $\mathbf{L} =$

$$
\begin{array}{cc}
\begin{array}{ccccc} A & B & C & D & E \end{array} & \\
\begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array}
\end{array}
$$



- so $\mathbf{Le} = \mathbf{0}$, but $\mathbf{e^T e} \neq \mathbf{0}$, so this solution doesn't work!
- What do we do now?

Let's look at the second smallest eigenvalue

- Calculate second smallest eigenvalue and eigenvector, λ_2 and $\mathbf{x_2}$
- Minimize Rayleigh-Ritz quotient
- $min_{\mathbf{x} \perp \mathbf{x_1}} \frac{\mathbf{x^T L x}}{x^T x}$
- note $\mathbf{x_2^T x_1} = \mathbf{x_2^T e} = \mathbf{0}$

Spectral Graph Partitioning Algorithm

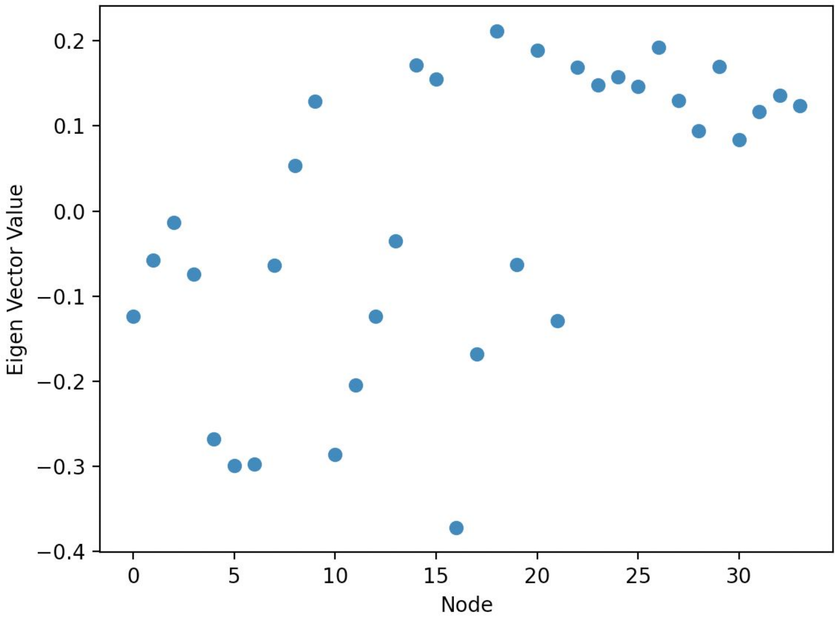
Algorithm 1 Spectral Graph Partitioning Algorithm

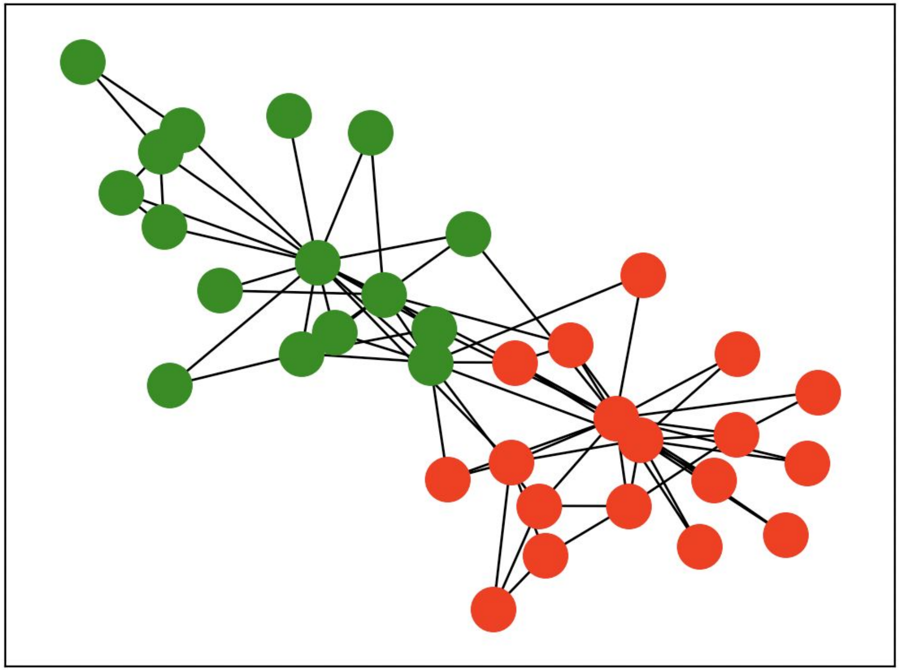
Input: $G = (V, E)$ and adjacency matrix A
Output: class indicator vector $\mathbf{s}$

compute $\mathbf{L} = \mathbf{D} - \mathbf{A}$
compute second smallest eigenvector
for min cut: solve $\mathbf{Lx} = \lambda\mathbf{x}$
for normalized cut: solve $\mathbf{Lx} = \lambda\mathbf{Dx}$
$\mathbf{s} = \mathbf{sign}(\mathbf{x_2})$

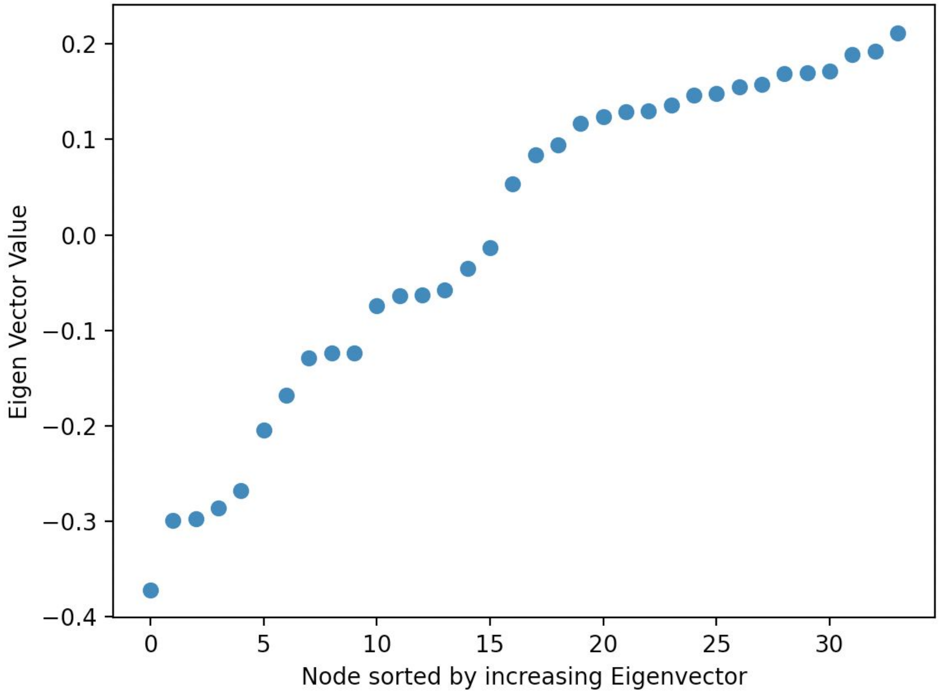
But this gives us only 2 partitions - need to do iteratively!

Spectral Partitioning for Zachary Karate Club



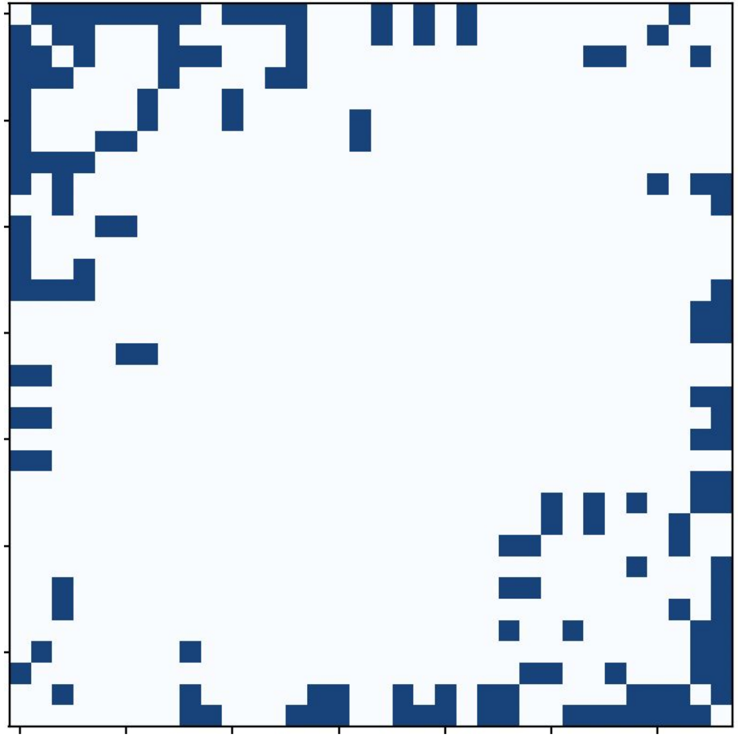


Spectral Partitioning for Zachary Karate Club

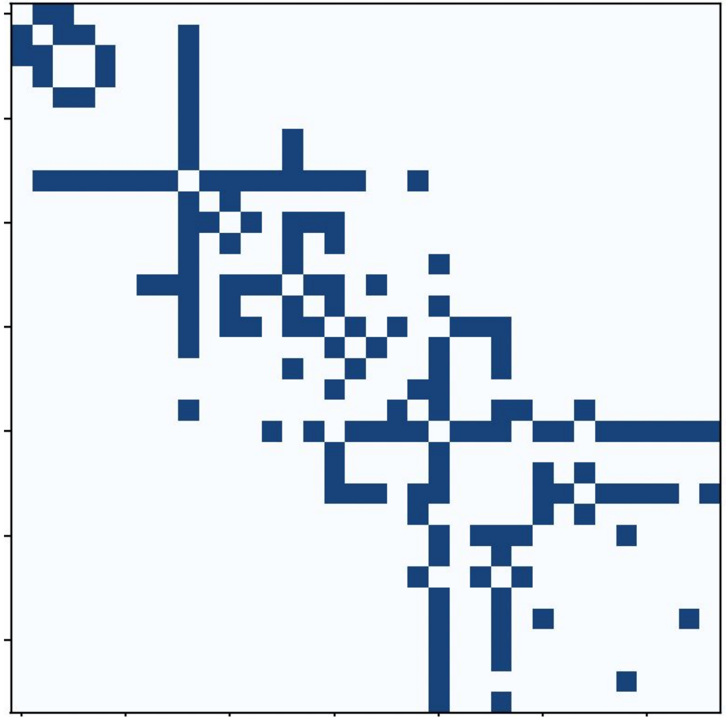


- Can look for jumps in the values
- These intuitively represent sub clusters or communities

Adjacency Matrix For Zachary Karate Club



Original Adjacency Matrix



After Sorting by Fiedler Vector