A First Course in Network Theory

Spectral Graph Partitioning

Luce le Gorrec, Philip Knight, Francesca Arrigo

University of Strathclyde, Glasgow

Graph Partitioning

<u>Preliminaries</u> Each graph $G = (V, E, \omega)$ is **non-directed**, **anti-reflexive**, *and with $\omega : E \to \mathbb{R}_+$. Also, $\forall u, v \in V$, $u \sim v$ means that $\{u, v\} \in E$.*

What ?
Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.

Graph Partitioning

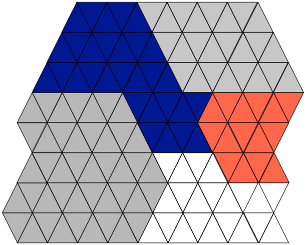
Preliminaries *Each graph* $G = (V, E, \omega)$ *is* **non-directed**, **anti-reflexive**, *and with* $\omega : E \to \mathbb{R}_+$*. Also,* $\forall u, v \in V$*,* $u \sim v$ *means that* $\{u, v\} \in E$*.*

What ?

Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.

What for ?

- **Efficiently solving PDEs**,
- Circuit boards in VLSI,
- Communities in social networks,
- Parallel/distributed computation,
- Etc.



Graph Partitioning

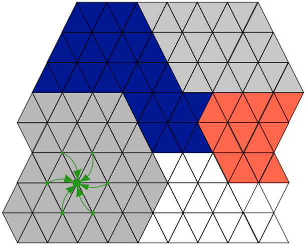
<u>*Preliminaries*</u> *Each graph $G = (V, E, \omega)$ is* **non-directed**, **anti-reflexive**, *and with $\omega : E \rightarrow \mathbb{R}_+$. Also, $\forall u, v \in V$, $u \sim v$ means that $\{u, v\} \in E$.*

What ?

Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.

What for ?

- **Efficiently solving PDEs**,
- Circuit boards in VLSI,
- Communities in social networks,
- Parallel/distributed computation,
- Etc.



Graph Partitioning

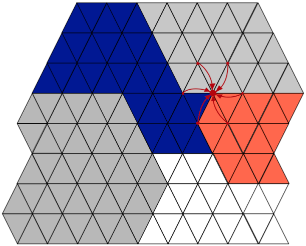
Preliminaries *Each graph* $G = (V, E, \omega)$ *is* **non-directed**, **anti-reflexive**, *and with* $\omega : E \to \mathbb{R}_+$*. Also,* $\forall u, v \in V$*,* $u \sim v$ *means that* $\{u, v\} \in E$*.*

What ?

Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.

What for ?

- **Efficiently solving PDEs**,
- Circuit boards in VLSI,
- Communities in social networks,
- Parallel/distributed computation,
- Etc.

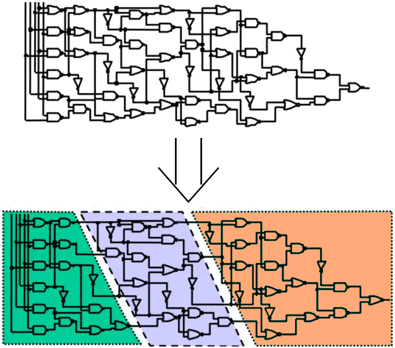


Graph Partitioning

<u>*Preliminaries*</u> *Each graph* $G = (V, E, \omega)$ *is* **non-directed**, **anti-reflexive**, *and with* $\omega : E \to \mathbb{R}_+$. *Also,* $\forall u, v \in V$, $u \sim v$ *means that* $\{u, v\} \in E$.

What ?

Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.



What for ?

- Efficiently solving PDEs,
- **Circuit boards in VLSI**,
- Communities in social networks,
- Parallel/distributed computation,
- Etc.

Graph Partitioning

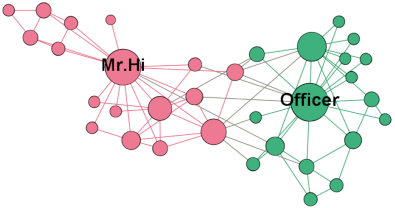
<u>*Preliminaries*</u> *Each graph* $G = (V, E, \omega)$ *is* **non-directed**, **anti-reflexive**, *and with* $\omega : E \to \mathbb{R}_+$. *Also,* $\forall u, v \in V$, $u \sim v$ *means that* $\{u, v\} \in E$.

What ?

Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.

What for ?

- Efficiently solving PDEs,
- Circuit boards in VLSI,
- **Communities in social networks**,
- Parallel/distributed computation,
- Etc.



Graph Partitioning

<u>Preliminaries</u> Each graph $G = (V, E, \omega)$ *is* **non-directed**, **anti-reflexive**, *and with* $\omega : E \rightarrow \mathbb{R}_+$. *Also,* $\forall u, v \in V$, $u \sim v$ *means that* $\{u, v\} \in E$.

What ?

Partitioning a graph $G = (V, E, \omega) \iff$ Partitioning its vertex set V.

What for ?

- Efficiently solving PDEs,
- Circuit boards in VLSI,
- Communities in social networks,
- **Parallel/distributed computation,**
- Etc.

Graph Partitioning – Cut

Roughly, partitioning a graph means finding the “almost disconnected” subsets of nodes within the network.

<u>Definition</u> Given $S \subset V$, **the weight of the cut induced by** S is

$$Cut(S) = \sum_{i \in S} \sum_{\substack{j \notin S: \\ i \sim j}} \omega(i,j).$$

<u>Examples</u>

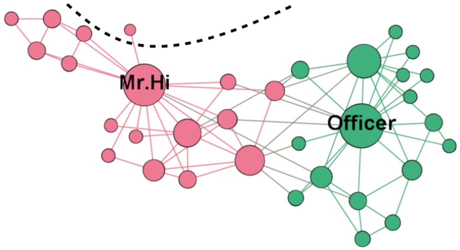
Graph Partitioning – Cut

Roughly, partitioning a graph means finding the "almost disconnected" subsets of nodes within the network.

Definition Given $S \subset V$, **the weight of the cut induced by S** is

$$Cut(S) = \sum_{i \in S} \sum_{\substack{j \notin S: \\ i \sim j}} \omega(i,j).$$

Examples



$\Longrightarrow$ When partitioning graphs, one wants to find a **non trivial set S** that induces a **lightly weighted cut**.

The Graph Laplacian – Relation with the Cut

Definition Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ the adjacency matrix of G, and $\mathbf{D} = diag(\mathbf{A1})$ its degree matrix, the **Laplacian of** G is the matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ such that $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

The Graph Laplacian – Relation with the Cut

Definition Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ the adjacency matrix of G, and $\mathbf{D} = diag(\mathbf{A1})$ its degree matrix, the **Laplacian of** G is the matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ such that $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Property $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i \sim j} \omega(i,j)(x_i - x_j)^2$.

Proof

The Graph Laplacian – Relation with the Cut

Definition Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ the adjacency matrix of G, and $\mathbf{D} = diag(\mathbf{A1})$ its degree matrix, the **Laplacian of** G is the matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ such that $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Property $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i \sim j} \omega(i,j)(x_i - x_j)^2$.

Proof

Corollary Given $S \subset V$, and $\mathbf{x} \in \{-1, 1\}^n$ s.t. $\mathbf{x}(i) = \begin{cases} 1 \text{ if } i \in S, \\ -1 \text{ else} \end{cases}$, namely **the signed indicator of** S, thus

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = 4 \times Cut(S).$$

Proof Exercise

The Graph Laplacian – Relation with the Cut

Definition Given $\mathbf{A} \in \mathbb{R}^{n\times n}$ the adjacency matrix of G, and $\mathbf{D} = diag(\mathbf{A1})$ its degree matrix, the **Laplacian of** G is the matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ such that $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Property $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{i\sim j} \omega(i,j)(x_i - x_j)^2$.

Proof

Corollary Given $S \subset V$, and $\mathbf{x} \in \{-1, 1\}^n$ s.t. $\mathbf{x}(i) = \begin{cases} 1 \text{ if } i \in S, \\ -1 \text{ else} \end{cases}$, namely **the signed indicator of** S, thus

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = 4 \times Cut(S).$$

Proof Exercise

The Balanced Cut Problem Given this corollary, finding the signed indicator of $S \subset V$ s.t. $|S| = n/2$ which induces a cut of minimum weight, can be written:

$$\begin{aligned} \text{minimise} \quad & \mathbf{x}^T\mathbf{L}\mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \{-1, 1\}^n, \\ \text{and} \quad & \mathbf{1}^T\mathbf{x} = 0. \end{aligned}$$

The Graph Laplacian – Bounding the Cut

Property The Laplacian $\mathbf{L}$ is positive semi definite, and $dim(Ker(\mathbf{L}))$ is the number of **connected components** in G.

Notation The sorted eigenvalues of $\mathbf{L}$, counted with multiplicity, are denoted $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

Proof

The Graph Laplacian – Bounding the Cut

Property The Laplacian **L** is positive semi definite, and $dim(Ker(\mathbf{L}))$ is the number of **connected components** in G.
Notation The sorted eigenvalues of **L**, counted with multiplicity, are denoted $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.
Proof

Property Given $S \subset V : |S| = n/2$, thus

$$\lambda_2 \times n/4 \leq Cut(S) \leq \lambda_n \times n/4.$$

Proof Exercise

The Graph Laplacian – Bounding the Cut

Property The Laplacian $\mathbf{L}$ is positive semi definite, and $dim(Ker(\mathbf{L}))$ is the number of **connected components** in G.
Notation The sorted eigenvalues of $\mathbf{L}$, counted with multiplicity, are denoted $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.
Proof

Property Given $S \subset V : |S| = n/2$, thus

$$\lambda_2 \times n/4 \leq Cut(S) \leq \lambda_n \times n/4.$$

Proof Exercise

Definition The eigenvalue λ_2 is called the **algebraic connectivity of** G.

Flavour The algebraic connectivity gives an indication about how close to disconnected the graph is.

Visual Illustration

Solving the Balanced Cut Problem

The Balanced Cut Problem of finding $S \subset V$ s.t. $|S| = n/2$ which induces a cut of minimum weight is equivalent to solve

$$\begin{aligned} \text{minimise} \quad & \mathbf{x}^T \mathbf{L} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \{-1, 1\}^n, \\ \text{and} \quad & \mathbf{1}^T \mathbf{x} = 0. \end{aligned}$$

which is NP-complete.

Solving the Balanced Cut Problem

The Balanced Cut Problem of finding $S \subset V$ s.t. $|S| = n/2$ which induces a cut of minimum weight is equivalent to solve

$$\begin{aligned} \text{minimise} \quad & \mathbf{x}^T \mathbf{L} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \{-1, 1\}^n, \\ \text{and} \quad & \mathbf{1}^T \mathbf{x} = 0. \end{aligned}$$

which is NP-complete.

$\implies$ Relaxing the 1st constraint: $\mathbf{x} \in \{-1, 1\}^n \rightsquigarrow \mathbf{x}^T \mathbf{x} = n$.

Solving the Balanced Cut Problem

The Balanced Cut Problem of finding $S \subset V$ s.t. $|S| = n/2$ which induces a cut of minimum weight is equivalent to solve

$$\begin{aligned} \text{minimise} \quad & \mathbf{x}^T \mathbf{L} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \{-1, 1\}^n, \\ \text{and} \quad & \mathbf{1}^T \mathbf{x} = 0. \end{aligned}$$

which is NP-complete.

$\implies$ Relaxing the 1st constraint: $\mathbf{x} \in \{-1, 1\}^n \rightsquigarrow \mathbf{x}^T \mathbf{x} = n$.

Property An eigenvector of norm $\sqrt{n}$ associated with λ_2 is always a solution of the Relaxed Balanced Cut Problem:

$$\begin{aligned} \text{minimise} \quad & \mathbf{x}^T \mathbf{L} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x}^T \mathbf{x} = n, \\ \text{and} \quad & \mathbf{1}^T \mathbf{x} = 0. \end{aligned}$$

Proof Exercise

Definition Such a vector is called the **Fiedler vector**, denoted by $\mathbf{x}_2$.

Balanced Cut Problem – Building S from the Fiedler vector

Median Threshold Find x_m the median of $\mathbf{x}_2$, and do

$$S = \{i \in V : \mathbf{x}(i) > x_m\} \cup \{\text{half of the } i's : \mathbf{x}_2(i) = x_m\}$$

Zero Threshold Do $S = \{i \in V : \mathbf{x}_2(i) > 0\}$

Guarantees about the connectivity of resulting subgraphs:

- Denoting $\overline{S} = V \setminus S$, $G_{\overline{S}} = (\overline{S}, E \cap (\overline{S} \times \overline{S}))$ is connected.
- If $\mathbf{x}_2(i) \neq 0, \forall i$, then $G_S = (S, E \cap (S \times S))$ is also connected.

Sweep Cut Method

```
[∼, inds] = sort(x2);
k0 ← −1; φ0 ← ∞;
for k = 1 : n do
    φcrt ← φ(inds(1 : k));
    if φcrt < φ0 then
        k0 ← k; φ0 ← φcrt;
    end if
end for
return S = inds(1 : k0);
```

where

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

Question: Why using φ (and not Cut)?

Balanced Cut Problem – Building S from the Fiedler vector

Median Threshold Find x_m the median of $\mathbf{x}_2$, and do

$$S = \{i \in V : \mathbf{x}(i) > x_m\} \cup \{\text{half of the } i's : \mathbf{x}_2(i) = x_m\}$$

Zero Threshold Do $S = \{i \in V : \mathbf{x}_2(i) > 0\}$
Guarantees about the connectivity of resulting subgraphs:

- Denoting $\overline{S} = V \setminus S,\ G_{\overline{S}} = (\overline{S}, E \cap (\overline{S} \times \overline{S}))$ is connected.
- If $\mathbf{x}_2(i) \neq 0, \forall i$, then $G_S = (S, E \cap (S \times S))$ is also connected.

Sweep Cut Method

```
[~, inds] = sort(x_2);
k_0 <- -1; φ_0 <- ∞;
for k = 1 : n do
    φ_crt <- φ(inds(1 : k));
    if φ_crt < φ_0 then
        k_0 <- k; φ_0 <- φ_crt;
    end if
end for
return S = inds(1 : k_0);
```

where

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

Question: Why using φ (and not Cut)?

Balanced Cut Problem – Building S from the Fiedler vector

Median Threshold Find x_m the median of $\mathbf{x}_2$, and do

$$S = \{i \in V : \mathbf{x}(i) > x_m\} \cup \{\text{half of the } i's : \mathbf{x}_2(i) = x_m\}$$

Zero Threshold Do $S = \{i \in V : \mathbf{x}_2(i) > 0\}$
Guarantees about the connectivity of resulting subgraphs:

- Denoting $\overline{S} = V \setminus S$, $G_{\overline{S}} = (\overline{S}, E \cap (\overline{S} \times \overline{S}))$ is connected.
- If $\mathbf{x}_2(i) \neq 0, \forall i$, then $G_S = (S, E \cap (S \times S))$ is also connected.

Sweep Cut Method

$[\sim, inds] = sort(\mathbf{x}_2)$;
$k_0 \leftarrow -1; \varphi_0 \leftarrow \infty$;
for $k = 1 : n$ **do**
  $\varphi_{crt} \leftarrow \varphi(inds(1:k))$;
  if $\varphi_{crt} < \varphi_0$ **then**
    $k_0 \leftarrow k; \varphi_0 \leftarrow \varphi_{crt}$;
  end if
end for
return $S = inds(1:k_0)$;

where

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

Question: Why using φ (and not Cut)?

The Cheeger's Inequality

Definition Given $S \subset V$, we denote by **S's isoperimetric ratio** the scalar

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

The Cheeger's Inequality

<u>Definition</u> Given $S \subset V$, we denote by **S's isoperimetric ratio** the scalar

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

<u>Theorem</u> The following bounds for the isoperimetric ratio are called **the Cheeger's inequality**:

$$\frac{\lambda_2}{2} \leq \min_{S \subset V} \varphi(S) \leq \sqrt{2\lambda_2 d^{\omega}_{max}}$$

<u>Proof</u>

The Cheeger's Inequality

Definition Given $S \subset V$, we denote by **S's isoperimetric ratio** the scalar

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

Theorem The following bounds for the isoperimetric ratio are called **the Cheeger's inequality**:

$$\frac{\lambda_2}{2} \leq \min_{S \subset V} \varphi(S) \leq \sqrt{2\lambda_2 d^{\omega}_{max}}$$

Proof

Corollary The subset $S^* \subset V$ returned by the sweep cut method verifies the Cheeger's inequality: $\frac{\lambda_2}{2} \leq \varphi(S^*) \leq \sqrt{2\lambda_2 d^{\omega}_{max}}$.

The Cheeger's Inequality

Definition Given $S \subset V$, we denote by **S's isoperimetric ratio** the scalar

$$\varphi(S) = \frac{Cut(S)}{min(|S|, n - |S|)}.$$

Theorem The following bounds for the isoperimetric ratio are called **the Cheeger's inequality**:

$$\frac{\lambda_2}{2} \leq \min_{S \subset V} \varphi(S) \leq \sqrt{2\lambda_2 d^{\omega}_{max}}$$

Proof

Corollary The subset $S^* \subset V$ returned by the sweep cut method verifies the Cheeger's inequality: $\frac{\lambda_2}{2} \leq \varphi(S^*) \leq \sqrt{2\lambda_2 d^{\omega}_{max}}$.

Property The isoperimetric ratio of S^* is at most $2 \times \sqrt{\min_{S \subset V} \varphi(S) d^{\omega}_{max}}$.

Proof Exercise

Take Home Messages

- **Partitioning a graph** onto two balanced[1] subsets, such that the cut is minimised, **is a NP-complete problem**.
- If we relax the discrete optimisation problem into a continuous one, the solution is the **Fiedler vector**, i.e. the eigenvector of the **Laplacian** associated with the **algebraic connectivity** (2nd smallest eigenvalue).
- The **isoperimetric ratio**, that measures the consistency of a partitioning, has its minimum bounded by **two inequalities that involve the algebraic connectivity**. These are called the **Cheeger's inequality**.
- The cut obtained by **applying the Sweep Cut Method to the Fiedler vector**, also verifies the Cheeger's inequality. Its isoperimetric ratio has **an upper bound that depends on the optimal solution**.

[1] The problem remains NP-complete when imbalance is allowed.