Snehal M. Shekatkar

Centre for modeling and simulation, S.P. Pune University, Pune Large-scale structure of complex networks (Part 2)

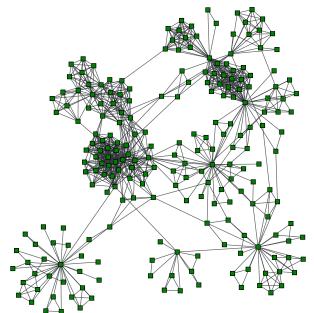
Large-scale structure of complex networks (Part 2)

Suehal M. Shekutkur

Centre for modeling and simulation,
SSP. Pane University, Pane

Hello

Community structure in networks





Large-scale structure of complex networks (Part 2) $\,$



Community structure in networks

Network of coauthorships in a university department

Community structure in networks

What are communities?

- ► **Traditional definition**: Groups of nodes with a high internal link density
- ▶ Modern definition: Nodes with similar connection probabilities to the rest of the network

Large-scale structure of complex networks (Part 2)

Community structure in networks

Community structure in networks

What are communities?

 Traditional definition: Groups of nodes with a high internal link density

> definition: Nodes with similar connectic ties to the rest of the network

Communities in the real-world networks

Social networks:

- ► Friend-circles
- ► Research communities
- Co-workers

World Wide Web:

- ▶ Pages with similar contents
- ▶ Webpages under the same domain (e.g. Wikipedia)

▶ Biological networks:

▶ Proteins with similar roles in protein interaction networks

4 D > 4 P > 4 E > 4 E > E 9 Q P

- Chemicals together taking part in chemical reactions in metabolic networks
- ▶ Communities in neuronal networks

Large-scale structure of complex networks (Part 2)

-Communities in the real-world networks

Communities in the real-world networks

- Friend-rirdes Research communities
- ▶ World Wide Web:
- · Pages with similar contents Webnares under the same domain (e.e. Wikinedia)
- · Proteins with similar roles in protein interaction networks
- · Chemicals together taking part in chemical reactions in metabolic networks Communities in neuronal networks

Community detection

Detecting communities is important!

- ▶ Communities are building blocks of networks
- ▶ Communities allow us to see "the big picture"
- ► Functional/Autonomous units
- ▶ Non-trivial effects on the processes on networks

Large-scale structure of complex networks (Part 2)

Community detection

2017-12-30

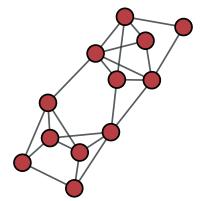
Community detection

Detecting communities is important!

- Communities are building blocks of netwo
- Communities allow us to see "the big pict
 - Functional/Autonomous units
- \blacktriangleright Non-trivial effects on the processes on networks

Graph partitioning

Problem of dividing a graph in a given number of groups of given sizes such that the number of links between the groups (cut size) is minimized



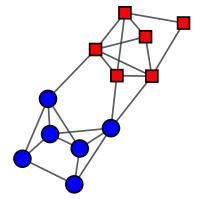
Large-scale structure of complex networks (Part 2) $\,$

└─Graph partitioning



Graph partitioning

Problem of dividing a graph in a given number of groups of given sizes such that the number of links between the groups (cut size) is minimized



Large-scale structure of complex networks (Part 2) $\,$

Problem of drivings a graph in a given number of groups of given sizes such that the number of flash between the groups (real size) in minimized

Graph partitioning

-Graph partitioning

Partitioning is hard!

- ightharpoonup Graph with n vertices
- ▶ Find two groups with sizes n_1 and n_2 such that the cut size is minimum
- Number of ways: $\frac{n!}{n_1!n_2!} \approx \frac{2^{n+1}}{\sqrt{n}}$

Heuristics are needed!

Large-scale structure of complex networks (Part 2)

Partitioning is hard!

Partitioning is hard!

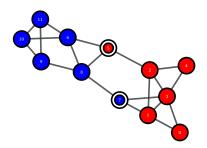
► Graph with n vertices

 Find two groups with sizes n₁ and n₂ such that the is minimum

 \blacktriangleright Number of ways: $\frac{n!}{n_1!n_2!}\approx \frac{2^{n+1}}{\sqrt{n}}$

Heuristics are needed

cut size = 4

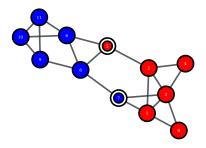


▶ Divide the vertices into two groups of the required sizes and calculate the cut size Large-scale structure of complex networks (Part 2)

└─Kernighan-Lin algorithm



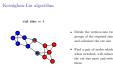
cut size = 4



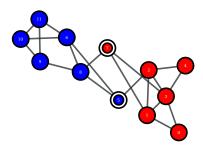
- ► Divide the vertices into two groups of the required sizes and calculate the cut size
- ► Find a pair of nodes which when switched, will reduce the cut size most and switch them

Large-scale structure of complex networks (Part 2)

└─Kernighan-Lin algorithm



cut size = 2



- ▶ Divide the vertices into two groups of the required sizes
- ► Find a pair of nodes which when switched, will reduce the cut size most and switch them

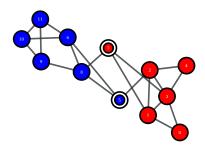
Large-scale structure of complex networks (Part 2)

—Kernighan-Lin algorithm

2017-12-30



cut size = 2



- ▶ Divide the vertices into two groups of the required sizes
- ► Find a pair of nodes which when switched, will reduce the cut size most and switch them
- ► If no such pair exists, select the pair which least increases the cut size

Large-scale structure of complex networks (Part 2)

-Kernighan-Lin algorithm

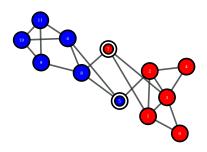
2017-12-30



Kernighan-Lin algorithm



cut size = 2

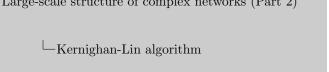


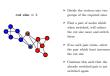
▶ Divide the vertices into two groups of the required sizes

2017-12-30

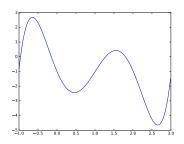
- ► Find a pair of nodes which when switched, will reduce the cut size most and switch them
- ► If no such pair exists, select the pair which least increases the cut size
- ▶ Continue this such that the already switched pair is not switched again

Large-scale structure of complex networks (Part 2)





Kernighan-Lin algorithm



- ▶ Go through all the states and select the one with the least cut size
- ▶ Start with this state and repeat the whole procedure
- ► Continue till the cut size no longer becomes smaller
- ► Starting with many random initial conditions is better

Large-scale structure of complex networks (Part 2)

-Kernighan-Lin algorithm

Kernighan-Lin algorithm

select the one with the least cut · Start with this state and repeat the whole procedure

Continue till the cut size no

► Go through all the states and

Group sizes remain constant

Spectral partitioning

- ▶ Faster algorithm than Kernighan-Lin
- ▶ Uses properties of the graph Laplacian
- ▶ More complex to implement than Kernighan-Lin

Large-scale structure of complex networks (Part 2)

_Spectral partitioning

Spectral partitioning

- ▶ Faster algorithm than Kernighan-Lin
- Uses properties of the graph Laplace
- \blacktriangleright More complex to implement than Kernighan-Lin

$$R = \frac{1}{2} \sum_{\substack{i,j \text{ in} \\ \text{different} \\ \text{groups}}} A_{ij}$$

Define

$$s_i = \begin{cases} +1 & \text{if vertex } i \text{ belongs to group 1} \\ -1 & \text{if vertex } i \text{ belongs to group 2} \end{cases}$$

Then

$$\frac{1}{2}(1 - s_i s_j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in different groups,} \\ 0 & \text{if } i \text{ and } j \text{ are in the same group} \end{cases}$$

First term,

$$\sum_{ij} A_{ij} = \sum_{i} k_i = \sum_{i} k_i s_i^2 = \sum_{ij} k_i \delta_{ij} s_i s_j$$

$$R = \frac{1}{4} \sum_{ij} (k_i \delta_{ij} - A_{ij}) s_i s_j = \frac{1}{4} \sum_{ij} L_{ij} s_i s_j$$

$$R = \frac{1}{4} \mathbf{s}^T \mathbf{L} \mathbf{s}$$

Large-scale structure of complex networks (Part 2)

2017-12-30

Spectral partitioning

Spectral partitioning
$$\begin{split} R &= \frac{1}{4} \sum_{ij} A_{ij} (1 - \kappa \nu_i) \end{split}$$
 First term, $\sum_{ij} A_{ij} &= \sum_{i} k_{ij} e^{i} - \sum_{ij} k_{i} d_{ij} \kappa \nu_i \\ R &= \frac{1}{4} \sum_{ij} (k_{i} d_{ij} - k_{ij}) e^{i} - \sum_{ij} k_{i} d_{ij} \kappa \nu_i \\ R &= \frac{1}{4} \sum_{ij} (k_{i} d_{ij} - k_{ij}) e^{i} - \sum_{ij} \sum_{k_{ij}} k_{ij} \kappa_i \end{split}$

L is so imp that we have a name for it! Laplacian

s is a column vector

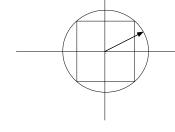
L: structure, s: division

find s that minimizes R

Problem is hard, s takes only integer values

Two constraints:

- \triangleright s_i can be only ± 1
- $\sum_{i} s_i = n_1 n_2 \Rightarrow \mathbf{1}^T \mathbf{s} = n_1 n_2$



Relax the first constraint

Large-scale structure of complex networks (Part 2)

Two constraints: s_i can be only ±1 \triangleright $\sum_{i} s_i = n_1 - n_2 \Rightarrow \mathbf{1}^T \mathbf{s} = n_1 - n_2$ Relax the first constraint

Relaxation method

-Relaxation method

hypercube

continuous s, differentiate

$$\frac{\partial}{\partial s_i} \left[\sum_{jk} L_{jk} s_j s_k + \lambda \left(n - \sum_j s_j^2 \right) + 2\mu \left((n_1 - n_2) - \sum_j s_j \right) \right] = 0$$

Large-scale structure of complex networks (Part 2)

_Spectral partitioning

 $\begin{aligned} & \text{Minimization with constraints} & + \text{Lagrange multipliers} \\ & \frac{\partial}{\partial s_i} \left[\sum_{j,k} L_{jk} s_j s_k + \lambda \left(n - \sum_j s_j^2 \right) + 2\mu \left((n_1 - n_2) - \sum_j s_j \right) \right] = 0 \end{aligned}$

Spectral partitioning

4 ロ ト 4 昼 ト 4 邑 ト 「邑 」 りなぐ

$$\frac{\partial}{\partial s_i} \left[\sum_{jk} L_{jk} s_j s_k + \lambda \left(n - \sum_j s_j^2 \right) + 2\mu \left((n_1 - n_2) - \sum_j s_j \right) \right] = 0$$

$$\sum_j L_{ij} s_j = \lambda s_i + \mu$$

Large-scale structure of complex networks (Part 2)

-Spectral partitioning

 $\sum L_{ij}s_j = \lambda s_i + \mu$

Spectral partitioning

$$\frac{\partial}{\partial s_i} \left[\sum_{jk} L_{jk} s_j s_k + \lambda \left(n - \sum_j s_j^2 \right) + 2\mu \left((n_1 - n_2) - \sum_j s_j \right) \right] = 0$$

$$\sum_j L_{ij} s_j = \lambda s_i + \mu$$

$$\mathbf{L}\mathbf{s} = \lambda \mathbf{s} + \mu \mathbf{1} = \lambda \left(\mathbf{s} + \frac{\mu}{\lambda} \mathbf{1} \right)$$

$$\mathbf{L} \left(\mathbf{s} + \frac{\mu}{\lambda} \mathbf{1} \right) = \lambda \left(\mathbf{s} + \frac{\mu}{\lambda} \mathbf{1} \right)$$

1 is an eigenvector of the Laplacian with eigenvalue 0

$$\mathbf{L}\mathbf{x} = \lambda \mathbf{x}$$



Large-scale structure of complex networks (Part 2)

```
-Spectral partitioning
```

Spectral partitioning $\begin{aligned} & \text{Minimization with constraints} & \to \text{Lagrange multipliers} \\ & \frac{\partial}{\partial v_i} \left[\sum_{jk} x_{jk} x_{jk} + \lambda \left(n - \sum_j x_j^2 \right) + 2\mu \left(n_1 - n_2 \right) - \sum_j x_j \right) \right] = 0 \\ & \sum_j x_{ij} x_{j} + \lambda x_i + \mu \\ & \sum_j x_{ij} x_{j} - \lambda x_i + \mu \\ & \text{La} & \to h + 2 1 - \lambda \left(n + \frac{\mu}{h} 1 \right) \\ & \text{L} \left(n + \frac{\mu}{h} 1 \right) - \lambda \left(n + \frac{\mu}{h} 1 \right) \\ & 1 \text{ is an eigenvector of the Lagrane with eigenvalue } 0 \end{aligned}$

 $\mathbf{L}\mathbf{x} = \lambda\mathbf{x}$

Spectral partitioning

 ${\bf x}$ is an eigenvector of the Laplacian with eigenvalue λ

Which eigenvector to choose?

$$\mathbf{x}$$
 cannot be the eigenvector $\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ . \\ . \\ 1 \end{pmatrix}$

$$\mathbf{1}^T \mathbf{x} = \mathbf{1} \left(\mathbf{s} + \frac{\mu}{\lambda} \mathbf{1} \right) = (n_1 - n_2) + \frac{\mu}{\lambda} n = 0$$

Large-scale structure of complex networks (Part 2)

Spectral partitioning

Spectral partitioning

al partitioning
$$\begin{split} \mathbf{x} &\text{ is an eigenvector of the Laplacian with eigenvalue } \lambda \\ &\text{Which eigenvector to choose?} \\ &\mathbf{x} &\text{ cannot be the eigenvector } \mathbf{1} = \begin{pmatrix} 1\\1\\1\\1 \end{pmatrix}, \\ \mathbf{1}^T\mathbf{x} &= \mathbf{1} \left(\mathbf{a} + \frac{\pi}{\lambda}\mathbf{1}\right) = (a_1 - a_2) + \frac{\pi}{\lambda}\mathbf{a} = 0 \end{split}$$

x is orthogonal to 1

x is eigenvector but not 1

Spectral partitioning

Which eigenvector to choose?

$$R = \frac{1}{4}\mathbf{s}^T \mathbf{L}\mathbf{s} = \frac{1}{4}\mathbf{x}^T \mathbf{x} = \frac{n_1 n_2}{n}\lambda$$

Choose the eigenvector with smallest possible eigenvalue!

Eigenvalues of the Laplacian are non-negative and smallest is always 0

 $\mathbf{v}_1 = \mathbf{1}$ is ruled out already. So choose \mathbf{v}_2 with the smallest positive eigenvalue

Large-scale structure of complex networks (Part 2)

-Spectral partitioning

Spectral partitioning

Which eigenvector to choose: $R = \frac{1}{s} \mathbf{r}^{T} \mathbf{L} \mathbf{s} = \frac{1}{s} \mathbf{x}^{T} \mathbf{x} = \frac{n_{1} n_{2}}{s} \lambda$

Eigenvalues of the Laplacian are non-negative and smallest it

 $v_1 = 1$ is ruled out already. So choose v_2 with the smallest positive eigenvalue

-Spectral partitioning

Thus, we want $\mathbf x$ to be as close as possible to $\mathbf s$

$$\mathbf{s} = x + \frac{n_1 - n_2}{n} \mathbf{1}$$

OR

$$s_i = x_i + \frac{n_1 - n_2}{n}$$

But s_i can be only ± 1

Thus, we want \mathbf{x} to be as close as possible to \mathbf{s}

Maximize:

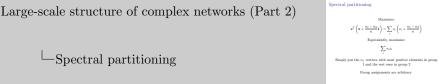
$$\mathbf{s}^T \left(\mathbf{x} + \frac{n_1 - n_2}{n} \mathbf{1} \right) = \sum_i s_i \left(x_i + \frac{n_1 - n_2}{n} \right)$$

Equivalently, maximize:

$$\sum_{i} s_i x_i$$

Simply put the n_1 vertices with most positive elements in group 1 and the rest ones in group 2

Group assignments are arbitrary



Choose the assignment with the smaller cut size

Spectral partitioning

- ightharpoonup Calculate \mathbf{v}_2 of the Laplacian
- ▶ Put vertices corresponding to largest n_1 elements in group 1 and others in group 2. Calculate the cut size
- ▶ Put vertices corresponding to smallest n_1 elements in group 1 and others in group 2. Calculate the cut size
- ► Choose the division with the smallest cut size among the two

Large-scale structure of complex networks (Part 2)

_Spectral partitioning

-

Spectral partitioning

 \blacktriangleright Calculate \mathbf{v}_2 of the Laplacian

 Put vertices corresponding to largest n₁ elements in group 1 and others in group 2. Calculate the cut size

 Put vertices corresponding to smallest n₁ elements in group 1 and others in group 2. Calculate the cut size

1 and others in group 2. Calculate the cut size

 Choose the division with the smallest cut size among the two

Community detection is harder!

► Graph partitioning

- ▶ well defined
- ▶ Number of groups is fixed
- ▶ Sizes of the groups are fixed
- ▶ Divide even if no good division exists

► Community detection

- ▶ ill-defined
- ▶ Number of groups depends on the structure of the network
- ▶ Sizes of the groups depend on the structure of the network
- ▶ Discover natural fault lines

Large-scale structure of complex networks (Part 2)

 \sqsubseteq Community detection is harder!

7-12-30

Community detection is harder!

▶ Graph partitioning

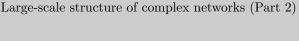
- well defined
- Number of groups is fixed
 Sizes of the groups are fixed
- Divide even if no good division exists

► Community detection

- Number of groups depends on the structure of the network
- Sizes of the groups depend on the structure of the n
- Discover natural fault lines

Many definitions.. many algorithms!

- ► Girvan-Newman algorithm
- ► Kernighan-Lin-Newman algorithm
- ► Spectral decomposition
- ► Clique-percolation
- ▶ Radom walk methods
- ► Statistical inference
- ► Label propagation
- ► Hierarchical clustering



2017-12-30

—Many definitions.. many algorithms!

Many definitions.. many algorithms!

- ► Girvan-Newman algorithm
- ► Kernighan-Lin-Newman algorithm
- ▶ Spectral decomposition
- ► Clique-percolation
- Radom walk method
- Statistical inference
- ► Label propagation
- ► Hierarchical clustering

I can go on.. These algorithms use different definitions/views of communities

Broad classification

► Agglomerative algorithms:

- ▶ Hierarchical clustering
- ▶ Louvain method
- ▶ CNM algorithm

▶ Divisive algorithms:

- ► Girvan-Newman algorithm
- ▶ Radichhi algorithm

► Assignment algorithms:

- ► Label propagation
- ► Spectral partitioning
- ► Kernighan-Lin-Newman algorithm

Large-scale structure of complex networks (Part 2)

-Broad classification

2017-12-30

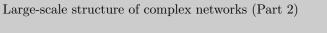
Broad classification

► Agglomerative algorithms:

- Hierarchical clustering
 Lorenia method
- Louvain method
 CNM algorithm
- Divisive algorithms:
 Girvan-Newman algorithm
- Girvan-Newman algorithm
 Radichhi algorithm
- Radichhi algorithm
 Assignment algorithms
- Label propagation
 Spectral partitioning
- Kernighan-Lin-Newman algorithm

"The" simplest community detection problem

- ightharpoonup Bisecting a graph with n nodes
- ► Group sizes are not fixed
- ▶ Minimum cut size?



 \sqsubseteq "The" simplest community detection problem

"The" simplest community detection problem

▶ Bisecting a graph with n nodes
 ▶ Group sizes are not fixed

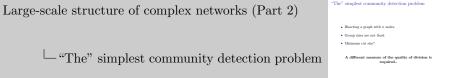
Group sizes are not fixed
 Minimum cut size?

Empty group

"The" simplest community detection problem

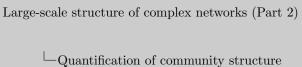
- ightharpoonup Bisecting a graph with n nodes
- ▶ Group sizes are not fixed
- ▶ Minimum cut size?

A different measure of the quality of division is required..



Different measure

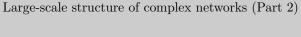
▶ Fewer than expected edges between the groups



Quantification of community structure $\mbox{ {\bf .}} \mbox{ {\bf .}} \mb$

few edges = expected edges = not a good division

- ▶ Fewer than expected edges between the groups
- ▶ Equivalently, more than expected edges inside the groups



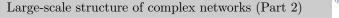
—Quantification of community structure

Force than espected edge between the propos
 Epirolastry, more than expected edges inside the gro

Remember assortativity

Quantification of community structure

- ► Fewer than expected edges between the groups
- ▶ Equivalently, more than expected edges inside the groups
- ► Assortativity mixing and modularity



└─Quantification of community structure

Quantification of community structure

Fewer than expected edges between the groups

Equivalently, more than expected edges inside the
 Assorbativity mixing and modularity

Assaultiny mang and me

Divide network using modularity

- ► Fewer than expected edges between the groups
- ▶ Equivalently, more than expected edges inside the groups
- ► Assortativity mixing and modularity
- ▶ Look for divisions with high modularity

Large-scale structure of complex networks (Part 2)

-Quantification of community structure

Quantification of community structure

- er than expected edges between the groups
- Equivalently, more than expected edges inside the g
 - Assortativity mixing and modu
 - Look for divisions with hi

Heuristics are needed

2017-12-30

- ▶ Fewer than expected edges between the groups
- ▶ Equivalently, more than expected edges inside the groups
- ► Assortativity mixing and modularity
- ▶ Look for divisions with high modularity
- ▶ Modularity maximization is hard

Large-scale structure of complex networks (Part 2)

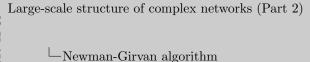
-Quantification of community structure

2017-12-30

Quantification of community structure

Newman-Girvan algorithm

- ► Look for edges between the communities
- ► Edge betweenness



Newman-Girvan algorithm

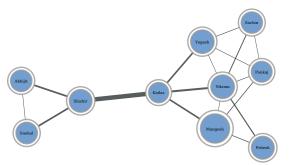
Look for edges between the communities

► Edge betweenness

Let's have a look at the edge betweenness

Edge betweenness

- ▶ Path between two nodes
- ► Shortest path between two nodes
- ▶ Number of shortest paths that go through a given edge



Large-scale structure of complex networks (Part 2)

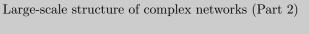
- Path between the structure of complex networks (Part 2)

- Path between the structure of the structure of



The algorithm

- ▶ Calculate betweenness for all edges
- ▶ Remove the edge with the highest betweenness
- ▶ Recalculate betweenness for all edges
- ► Repeat







- ► Calculate betweenness for all edges
- Remove the edge with the highest between
- ➤ Recalculate betweenness for all edges
 - peat