# Large-scale structure of complex networks (Part 2)

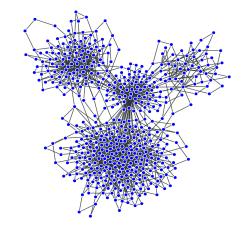
Snehal M. Shekatkar

Centre for modeling and simulation, S.P. Pune University, Pune

Hello

Large-scale structure of complex networks (Part 2)

# Community structure in networks

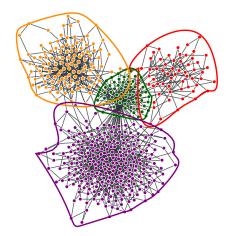


Large-scale structure of complex networks (Part 2)

Community structure in networks



# Community structure in networks



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
- ▶ Modern definition: Nodes with similar connection probabilities to the rest of the network

Large-scale structure of complex networks (Part 2)

2018-01-06

-Community structure in networks

Community structure in networks

What are communities?

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

### 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 A > 4 B > 4 B > B 9 Q (>

- 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

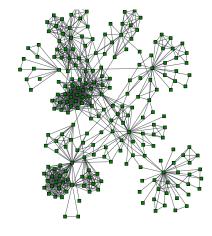
Communities in the real-world networks

Social networks:

- Friend-circles
   Research communities
- Co-workers
   World Wide Web:
- Pages with similar contents
   Webpages under the same domain (e.g. Wikipedia)
- 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)

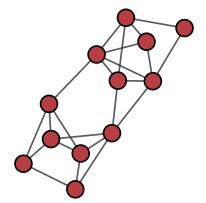


Community detection

Community detection

### 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

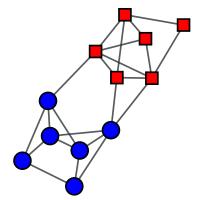
2018-01-06



4□▶ 4□▶ 4 亘 ▶ 4 亘 ▶ 9 Q (や)

### 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 dividing a graph in a given number of groups of given sizes such that the number of flash between the groups (cast size) is minimized.

Graph partitioning

-Graph partitioning

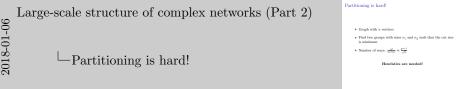
2018-01-06



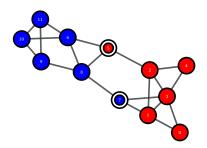
### Partitioning is hard!

- 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!



cut size = 4

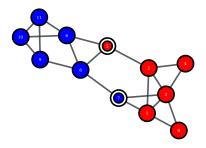


▶ Divide the vertices into two groups of the required sizes and calculate the cut size Large-scale structure of complex networks (Part 2)

└─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

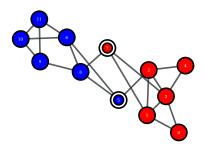
Large-scale structure of complex networks (Part 2)

└─Kernighan-Lin algorithm



Kernighan-Lin algorithm

cut size = 2



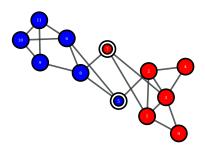
- ▶ Divide the vertices into two groups of the required sizes
- ► Find a pair of nodes which when switched, will reduce the cut size most and switch them

Large-scale structure of complex networks (Part 2)

60-10-80-10-



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

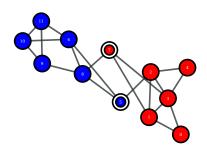
2018-01-06



Kernighan-Lin algorithm



cut size = 2



▶ Divide the vertices into two groups of the required sizes

2018-01-06

- ► 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)

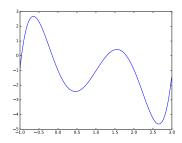
-Kernighan-Lin algorithm



Kernighan-Lin algorithm

If no such pair exists, select the pair which least increases

· Continue this such that the already switched pair is not



- ▶ 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

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

-Kernighan-Lin algorithm

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 Lapla
- $\blacktriangleright$  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

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

Spectral partitioning  $R = \frac{1}{2} \sum_{\substack{i \in A_i \text{ odd} \\ \text{defined}}} A_{ij}$  Define  $s_i = \begin{cases} -1 & \text{if were it binage to group 1} \\ -1 & \text{if were it binage to group 2} \end{cases}$  Then  $\frac{1}{2} (1 - s_i s_i) = \begin{cases} 1 & \text{if if and } j \text{ on in different groups}, \\ 0 & \text{if it and } j \text{ are in different groups}, \end{cases}$ 

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

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)

\_\_Spectral partitioning

Spectral partitioning 
$$\begin{split} R &= \frac{1}{4} \sum_{ij} A_{ij} (1-\kappa s_i) \end{split}$$
 First term,  $\sum_{ij} A_{ij} &= \sum_{i} b_i c_i^2 = \sum_{ij} b_i b_{ij} \kappa s_j \\ R &= \frac{1}{2} \sum_{ij} (b_i d_{ij} - A_{ij}) \kappa s_j = \frac{1}{2} \sum_{ij} L_{ij} \kappa s_j \\ R &= \frac{1}{4} \sum_{ij} (b_i d_{ij} - A_{ij}) \kappa s_j = \frac{1}{4} \sum_{ij} L_{ij} \kappa s_j \end{split}$ 

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

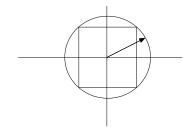
s is a column vector

L: structure, s: division

find s that minimizes R

Problem is hard, s takes only integer values

$$\triangleright$$
  $s_i$  can be only  $\pm 1$ 



Relax the first constraint

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

Two constraints:  $\bullet \ s_i \ \text{can be only} \ \exists 1$   $\bullet \ \sum_i s_i = n_1 - n_2 \Rightarrow 1^T \mathfrak{a} = n_1 - n_2$  Relax the first constraint

Relaxation method

Relaxation method

hypercube

continuous s, differentiate

 $\frac{\partial}{\partial u}\left[\sum L_{jk}s_js_k + \lambda \left(n - \sum s_j^2\right) + 2\mu \left((n_1 - n_2) - \sum s_j\right)\right] = 0$ 

Minimization with constraints  $\Rightarrow$  Lagrange multipliers

Minimization with constraints 
$$\Rightarrow$$
 Lagrange multipliers

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

-Spectral partitioning

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

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

Large-scale structure of complex networks (Part 2)

-Spectral partitioning

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

Spectral partitioning

◆ロト ◆部ト ◆恵ト ◆恵ト ・恵 ・ 釣り(で)

-Spectral partitioning

Spectral partitioning

 $L\left(s + \frac{\mu}{2}\mathbf{1}\right) = \lambda\left(s + \frac{\mu}{2}\mathbf{1}\right)$ 

 $Lx = \lambda x$ 

Minimization with constraints  $\Rightarrow$  Lagrange multipliers

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

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

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

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

1 is an eigenvector of the Laplacian with eigenvalue 0

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

# Spectral partitioning

 $\mathbf{x}$  is an eigenvector of the Laplacian with eigenvalue  $\lambda$ 

Which eigenvector to choose?

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

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

Spectral partitioning Large-scale structure of complex networks (Part 2) -Spectral partitioning  $\mathbf{1}^{T}\mathbf{x} = \mathbf{1}\left(\mathbf{s} + \frac{\mu}{\lambda}\mathbf{1}\right) = (n_{1} - n_{2}) + \frac{\mu}{\lambda}n = 0$ 

x is orthogonal to 1

x is eigenvector but not 1

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

◆ロト ◆部ト ◆恵ト ◆恵ト ・恵 ・ 釣り(で)

### 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?  $R = \frac{1}{2}\mathbf{s}^{T}\mathbf{L}\mathbf{s} = \frac{1}{2}\mathbf{x}^{T}\mathbf{x} = \frac{n_{1}n_{2}}{2}\lambda$ 

Choose the eigenvector with smallest possible eigenvalue! Eigenvalues of the Laplacian are non-negative and smallest i

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

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

### $\mathbf{OR}$

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

But  $s_i$  can be only  $\pm 1$ 

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

# Spectral partitioning

Maximize:

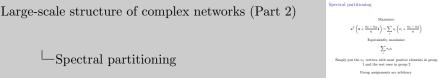
$$\mathbf{s}^T \left( \mathbf{x} + \frac{n_1 - n_2}{n} \mathbf{1} \right) = \sum_i s_i \left( x_i + \frac{n_1 - n_2}{n} \right)$$

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 n<sub>1</sub> elements in group
- Put vertices corresponding to smallest n<sub>1</sub> elements in group 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)

-Community detection is harder!

2018-01-06

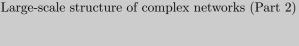
Community detection is harder!

#### · Graph partitioning

- well defined
- Number of groups is fixed · Sizes of the groups are fixed Divide even if no rood division exists
- ► Community detection
- · Number of groups depends on the structure of the network
  - · 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



2018-01-06

Many definitions.. many algorithms! Label propagation

Many definitions.. many algorithms!

► Kernighan-Lin-Newman algorithm

▶ Spectral decomposition

Clique-percolation

Radom walk method

Statistical inference

Hierarchical clustering

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

### Broad classification

### ► Agglomerative algorithms:

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

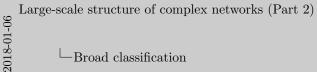
### ▶ Divisive algorithms:

- ► Girvan-Newman algorithm
- ▶ Radichhi algorithm

#### ► Assignment algorithms:

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

◆ロト ◆部ト ◆恵ト ◆恵ト ・恵 ・ 釣り(で)



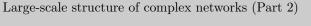
#### Broad classification

► Agglomerative algorithms:

- Hierarchical clustering
- Louvain method
   CNM algorithm
- ▶ Divisive algorithms:
- Girvan-Newman algorit
   Destination of the control of the c
- Radichhi algorithm
   Assignment algorithms
- Label propagation
   Sported 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?
  - ► Trivial partition
  - ▶ Needs ad hoc specification of sizes



— "The" simplest community detection problem

"The" simplest community detection problem

• Historing a graph with n moles

• Group sines are not fixed

• Minimum cut size?

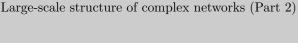
Empty group

2018-01-06

### "The" simplest community detection problem

- $\triangleright$  Bisecting a graph with n nodes
- ► Group sizes are not fixed
- ► Minimum cut size?
  - ► Trivial partition
  - ▶ Needs ad hoc specification of sizes

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





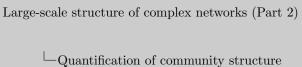


A different measure of the quality of division

Different measure

2018-01-06

▶ Fewer than expected edges between the groups

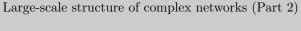


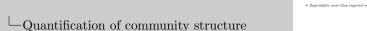
Quantification of community structure

Fewer than expected edges between the ground

few edges = expected edges = not a good division

- ► Fewer than expected edges between the groups
- $\blacktriangleright$  Equivalently, more than expected edges inside the groups



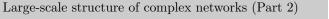


Remember assortativity

Quantification of community structure

 $\blacktriangleright$  Fewer than expected edges between the groups

- ► 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
- Assortativity mixing and modularity

Divide network using modularity

2018-01-06

- ► 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

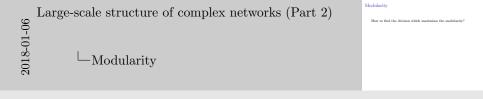
Quantification of community structure

- ewer than expected edges between the groups
- Equivalently, more than expected edges inside the
- Assortativity mixing and modulari
- Look for divisions with migh meanuari

Heuristics are needed

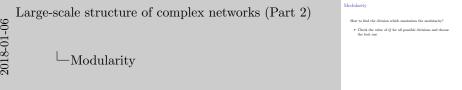
2018-01-06

How to find the division which maximizes the modularity?



How to find the division which maximizes the modularity?

lacktriangle Check the value of Q for all possible divisions and choose the best one



How to find the division which maximizes the modularity?

- lacktriangle Check the value of Q for all possible divisions and choose the best one
- Consider, N = 100,  $n_1 = n_2 = 50$

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

How to find the division which maximizes the modularity? • Check the value of Q for all possible divisions and choose the bost one • Consider, N=100,  $n_1=n_2=50$ 

Modularity

└─Modularity

How to find the division which maximizes the modularity?

- ightharpoonup Check the value of Q for all possible divisions and choose the best one
- Consider, N = 100,  $n_1 = n_2 = 50$
- ▶ Total possible divisions =  $^{100}C_{50} > 10^{29}$

Large-scale structure of complex networks (Part 2)

└─Modularity

Modularity

► Check the value of O for all possible divisions and choose

How to find the division which maximizes the modularity?

- lacktriangle Check the value of Q for all possible divisions and choose the best one
- Consider, N = 100,  $n_1 = n_2 = 50$
- ▶ Total possible divisions =  $^{100}C_{50} > 10^{29}$
- ▶ With a fast computer which checks 100 billion divisions per second:  $3 \times 10^{10}$  years!

Large-scale structure of complex networks (Part 2)

 $\sqsubseteq$  Modularity

#### Modularity

How to find the division which maximizes the modularity?

 Check the value of Q for all possible divisions and choose the best one

- $\blacktriangleright$  Consider,  $N=100,\,n_1=n_2=50$
- Total possible divisions = <sup>100</sup>C<sub>50</sub>
- ➤ With a fast computer which checks 100 billion divisions per cocond, 2 × 1010 month.

How to find the division which maximizes the modularity?

- ightharpoonup Check the value of Q for all possible divisions and choose the best one
- Consider, N = 100,  $n_1 = n_2 = 50$
- ▶ Total possible divisions =  $^{100}C_{50} > 10^{29}$
- ▶ With a fast computer which checks 100 billion divisions per second:  $3 \times 10^{10}$  years!
- ► Clever heuristics are required

Large-scale structure of complex networks (Part 2)

└─Modularity

Modularity

How to find the division which maximizes the modularity?

- Check the value of Q for all possible divisions and choose the best one

- $\blacktriangleright$  Consider,  $N=100,\,n_1=n_2=50$
- ▶ Total possible divisions =  $^{100}C_{50} > 1$
- With a fast computer which checks 100 billion divisions per second: 3 × 10<sup>10</sup> years?
- ▶ Clever heuristics are required

018-01

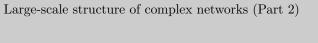
-Kernighan-Lin-Newman algorithm

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

▶ Start with a random division of the nodes



-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

Start with a random division of the nod

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

- ▶ Start with a random division of the nodes
- ► Change in modularity for shifting each vertex to the other group

Large-scale structure of complex networks (Part 2)

└─Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

 Change in modularity for shifting each vertex to the other group

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

- ▶ Start with a random division of the nodes
- ► Change in modularity for shifting each vertex to the other group
- ► Choose vertex whose shift makes maximum modularity change

Large-scale structure of complex networks (Part 2)

2018-01-06

—Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

 Change in modularity for shifting each vertex to the other group.

 Choose vertex whose shift makes maximum modularity change

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

◆ロト ◆御ト ◆恵ト ◆恵ト 恵 めらで

- ▶ Start with a random division of the nodes
- ► Change in modularity for shifting each vertex to the other group
- ► Choose vertex whose shift makes maximum modularity change
- ▶ If no such vertex exists, choose the one resulting in the least decrease in the modularity

Large-scale structure of complex networks (Part 2)

-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

 Change in modularity for shifting each vertex to the other group.

 Choose vertex whose shift makes maximum modularity change

 If no such vertex exists, choose the one resulting in the least decrease in the modularity

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

4□ > 4□ > 4□ > 4□ > 4□ > 4□ > 4□

- ▶ Start with a random division of the nodes
- ► Change in modularity for shifting each vertex to the other group
- ► Choose vertex whose shift makes maximum modularity change
- ▶ If no such vertex exists, choose the one resulting in the least decrease in the modularity
- ▶ Repeat so that the vertex once moved is not moved again

Large-scale structure of complex networks (Part 2)

L

-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

 Change in modularity for shifting each vertex to the other group.

 Choose vertex whose shift makes maximum modularity change

If no such vertex exists, choose the one resulting in the

 If no such vertex exists, choose the one resulting in least decrease in the modularity

 $\,\blacktriangleright\,$  Repeat so that the vertex once moved is not moved again

Repeat so that the vertex once moved is not moved ag

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

◆ロト ◆団ト ◆重ト ◆重ト ● のQで

- ▶ Start with a random division of the nodes
- ► Change in modularity for shifting each vertex to the other group
- ► Choose vertex whose shift makes maximum modularity change
- ▶ If no such vertex exists, choose the one resulting in the least decrease in the modularity
- ▶ Repeat so that the vertex once moved is not moved again
- ► Select a state with the highest modularity

Large-scale structure of complex networks (Part 2)

2018-01-06

-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

► Change in modularity for shifting each vertex to the other

 Choose vertex whose shift makes maximum modularity change

If no such vertex exists, choose the one resulting in the

 If no such vertex exists, choose the one resulting in least decrease in the modularity

Repeat so that the vertex once moved is not moved.

 ${\blacktriangleright}\,$  Select a state with the highest modularity

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

- ► Start with a random division of the nodes
- ► Change in modularity for shifting each vertex to the other group
- ► Choose vertex whose shift makes maximum modularity change
- ▶ If no such vertex exists, choose the one resulting in the least decrease in the modularity
- ▶ Repeat so that the vertex once moved is not moved again
- ▶ Select a state with the highest modularity
- ▶ Repeat the whole process starting with this state till the modularity stabilizes

Large-scale structure of complex networks (Part 2)

-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

· Change in modularity for shifting each vertex to the other

· Choose vertex whose shift makes maximum modularity

· If no such vertex exists, choose the one resulting in the

· Select a state with the highest modularit

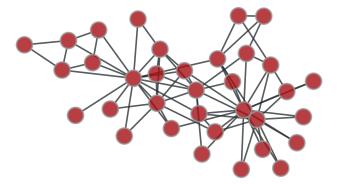
· Repeat the whole process starting with this state till the

Sizes of the groups are not fixed

No swapping

Variation of KL algorithm

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



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



Does somebody know this network?

# Zachry karate club network

10-10-01

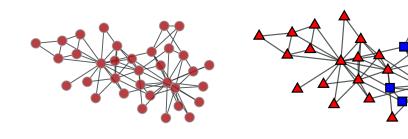
Large-scale structure of complex networks (Part 2)  $\,$ 



Zachry karate club network

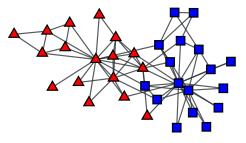
2018-01-06

└─Zachry karate club network



# Application to Zachry karate club

Actual division



Large-scale structure of complex networks (Part 2)

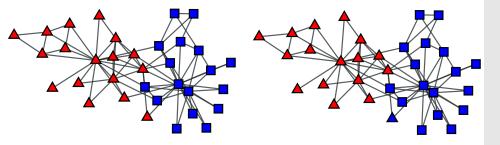
L—Application to Zachry karate club



# Application to Zachry karate club

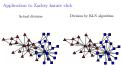
Actual division

Division by KLN algorithm



Large-scale structure of complex networks (Part 2)

\_\_Application to Zachry karate club



$$Q = \frac{1}{2m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) = \frac{1}{2m} \sum_{ij} B_{ij} \delta(c_i, c_j)$$

Note that:

$$\sum_{i} B_{j} = \sum_{i} A_{ij} - \frac{k_{i}}{2m} \sum_{i} k_{j} = k_{i} - \frac{k_{i}}{2m} 2m = 0$$

Large-scale structure of complex networks (Part 2)

-Spectral modularity maximization

 $Q = \frac{1}{2m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) = \frac{1}{2m} \sum_{ij} B_{ij} k(c_i, c_j)$  Note that:  $\sum B_j = \sum A_{ij} - \frac{k_i}{2m} \sum k_j - k_i - \frac{k_i}{2m} 2m = 0$ 

Spectral modularity maximization

spectral partitioning: cut size

analogous algorithm exists

# Spectral modularity maximization

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

Large-scale structure of complex networks (Part 2)



—Spectral modularity maximization

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

$$\frac{1}{2}(1+s_is_j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ belong to the same group} \\ 0 & \text{Otherwise} \end{cases}$$

Large-scale structure of complex networks (Part 2)

 $\sqsubseteq$  Spectral modularity maximization

 $s_i = \begin{cases} +1 & \text{if vertex $i$ belongs to group $1$} \\ -1 & \text{if vertex $i$ belongs to group $2$} \end{cases}$   $\frac{1}{2}(1+s_is_j) = \begin{cases} 1 & \text{if $i$ and $j$ belong to the same group $2$} \\ 0 & \text{Otherwise} \end{cases}$ 

Spectral modularity maximization

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

$$\frac{1}{2}(1+s_is_j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ belong to the same group} \\ 0 & \text{Otherwise} \end{cases}$$

$$B = \frac{1}{2m} \sum_{ij} B_{ij} \delta(c_i, c_j) = \frac{1}{4m} \sum_{ij} B_{ij} (1 + s_i s_j) = \frac{1}{4m} \sum_{ij} B_{ij} s_i s_j$$
$$Q = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

Large-scale structure of complex networks (Part 2)

-Spectral modularity maximization

$$\begin{split} s_1 &= \begin{cases} +1 & \text{if write i belongs to group 1} \\ -1 & \text{if write is belongs to group 2} \end{cases} \\ &= \frac{1}{2}(1+s_{s,t}) - \begin{cases} s_1 & \text{if and j belong to the same group} \end{cases} \\ &= \frac{1}{2m} \sum_{ij} R_{ij} \theta_{i(ij,t_j)} - \frac{1}{4m} \sum_{ij} R_{ij} (s_{ij}s_j) - \frac{1}{4m} \sum_{ij} R_{ij} s_{ij} s_j \\ &= \frac{1}{2m} \sum_{ij} R_{ij} \theta_{i(ij,t_j)} - \frac{1}{4m} \sum_{ij} R_{ij} s_{ij} s_j \end{cases} \end{split}$$

Spectral modularity maximization

### Spectral modularity maximization

#### Relaxation method

- ▶ Numbers of elements with values +1 and -1 are not fixed
- Only constraint:  $\mathbf{s}^T \mathbf{s} = \sum s_i^2 = n$

Only constraint: 
$$\mathbf{s}^2 \mathbf{s} = \sum_{i} s_i^2 = n$$

$$\frac{\partial}{\partial s_i} \left[ \sum_{ij} B_{jk} s_j s_k + \beta \left( n - \sum_{j} s_j^2 \right) \right] = 0$$

$$\sum_{j} B_{ij} s_j = \beta s_i$$

$$\mathbf{B}\mathbf{s} = \beta\mathbf{s}$$

Large-scale structure of complex networks (Part 2)

-Spectral modularity maximization



s is eigenvector of modularity matrix

Spectral modularity maximization

# Spectral modularity maximization

$$Q = \frac{1}{4m} \beta \mathbf{s}^T \mathbf{B} \mathbf{s} = \frac{1}{4m} \beta \mathbf{s}^T \mathbf{s} = \frac{n}{4m} \beta$$

Thus, choose  $\mathbf{s}$  to be the eigenvector  $\mathbf{u}_1$  corresponding to the largest eigenvalue of the modularity matrix Maximize:

$$\mathbf{s}^T \mathbf{u}_1 = \sum_i s_i u_{1i}$$

Maximum is achieved when each term is non-negative  $\Rightarrow$  Use signs of  $u_{1i}$ !

Large-scale structure of complex networks (Part 2)

—Spectral modularity maximization

Spectral modularity maximization

Spectral modularity maximization

 $Q = \frac{1}{4m}\beta s^T B s - \frac{1}{4m}\beta s^T s - \frac{n}{4m}\beta$  Thus, choose to be the eigenvector  $\mathbf{u}_1$  corresponding to largest eigenvalue of the modularity matrix Maximize  $\mathbf{s}^T \mathbf{u}_1 = \sum s_i \mathbf{u}_i.$ 

Maximum is achieved when each term is non-negative ⇒ Us

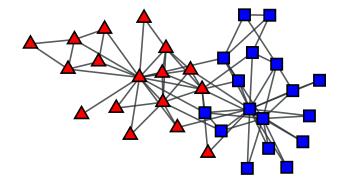
4D + 4B + 4B + B + 990

-Spectral modularity maximization

- ► Calculate the modularity matrix
- ► Calculate its eigenvector corresponding to the largest eigenvalue
- ▶ Assign nodes to communities based on the signs of elements

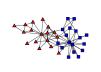
↓□▶ ←□▶ ←□▶ ←□▶ □ ♥♀♡

### Application to karate club network



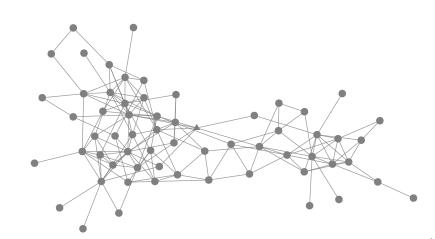
Large-scale structure of complex networks (Part 2)

\_\_Application to karate club network



Application to karate club network

# Bottlenose dolphins



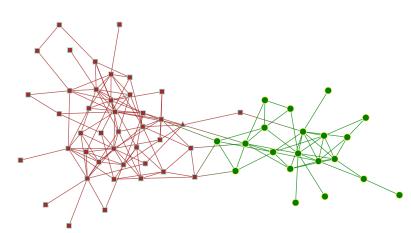
¹Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) Behav Ecol Sociobiol 54:396405

Large-scale structure of complex networks (Part 2)

Bottlenose dolphins



# Bottlenose dolphins

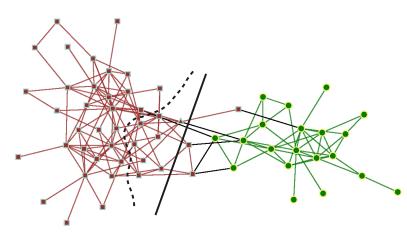


<sup>2</sup>Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) Behav Ecol Sociobiol 54:396405

Bottlenose dolphins Large-scale structure of complex networks (Part 2) -Bottlenose dolphins



# Bottlenose dolphins

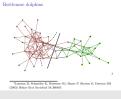


<sup>3</sup>Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM

(2003) Behav Ecol Sociobiol 54:396405

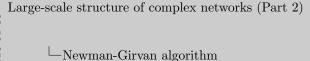
Large-scale structure of complex networks (Part 2)

Bottlenose dolphins



## Newman-Girvan algorithm

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



Newman-Girvan algorithm

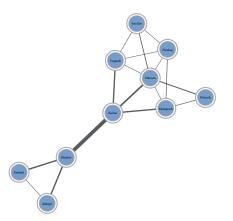
Look for edges between the communities

► 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



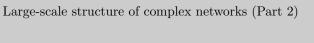
#### Large-scale structure of complex networks (Part 2)

—Edge betweenness



#### The algorithm

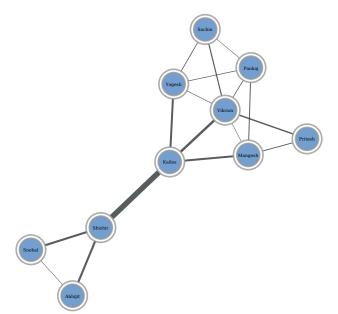
- ► Calculate betweenness for all edges
- ▶ Remove the edge with the highest betweenness
- ▶ Recalculate betweenness for all edges
- ► Repeat





The algorithm

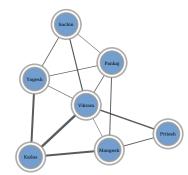
The algorithm

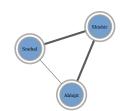


Large-scale structure of complex networks (Part 2)



└─Girvan-Newman algorithm



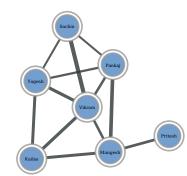


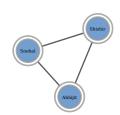
2018-01-06

Large-scale structure of complex networks (Part 2)

Givan-Newman algorithm

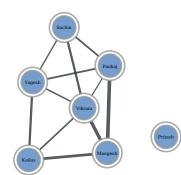
└─Girvan-Newman algorithm

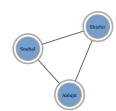




Large-scale structure of complex networks (Part 2)  $\begin{tabular}{l} $ \sqsubseteq$ Girvan-Newman algorithm \end{tabular}$ 







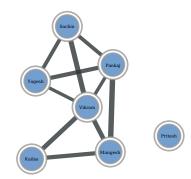
2018-01-06

Large-scale structure of complex networks (Part 2)

Girvan-Newman algorithm

Girvan-Newman algorithm



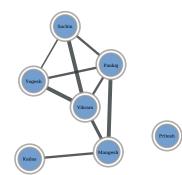


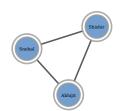


Large-scale structure of complex networks (Part 2)  $\begin{tabular}{l} $ \sqsubseteq $Girvan-Newman algorithm \end{tabular}$ 





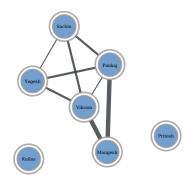


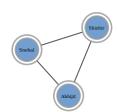


2018-01-06

Large-scale structure of complex networks (Part 2)



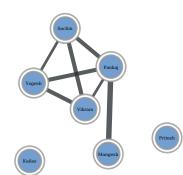






Large-scale structure of complex networks (Part 2)



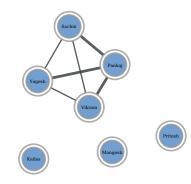


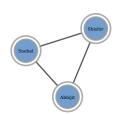




Large-scale structure of complex networks (Part 2)



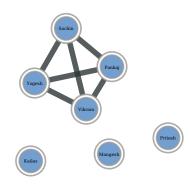




2018-01-06

Large-scale structure of complex networks (Part 2)

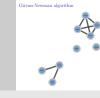
Girvan-Newman algorithm

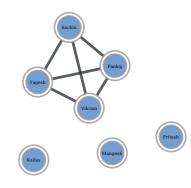


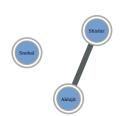




Large-scale structure of complex networks (Part 2)



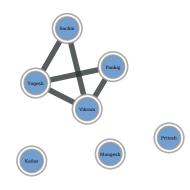


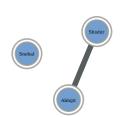


2018-01-06

Large-scale structure of complex networks (Part 2)

Givan-Newman algorithm

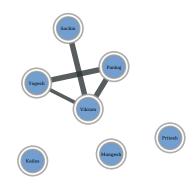


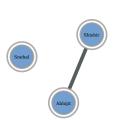


Large

Large-scale structure of complex networks (Part 2)

Givan-Neeman algorithm

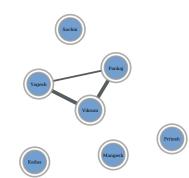




2018-01-06

Large-scale structure of complex networks (Part 2)

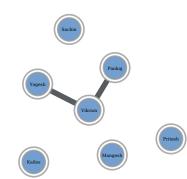




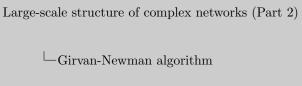


Large-scale structure of complex networks (Part 2)  $\begin{tabular}{l} $ \sqsubseteq $Girvan-Newman algorithm \end{tabular}$ 

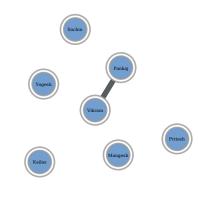


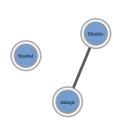


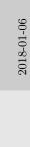


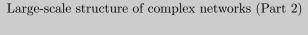






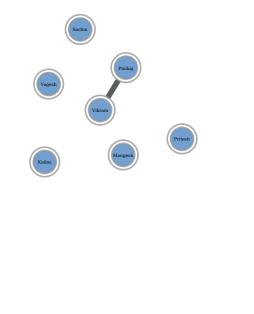








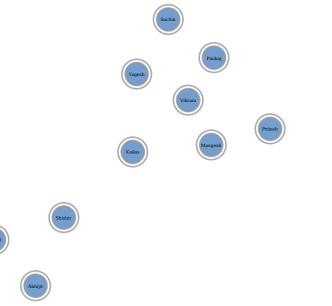
 $\stackrel{\&}{\sim}$   $\sqsubseteq$  Girvan-Newman algorithm

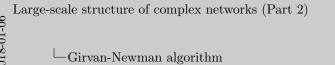


Large-scale structure of complex networks (Part 2)

Girvan-Newman algorithm

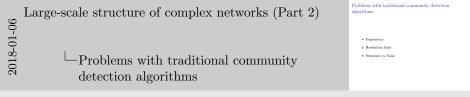




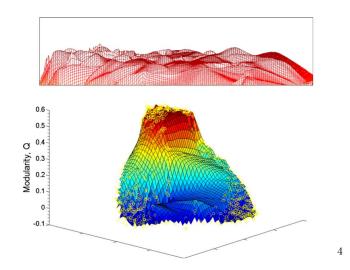




- ▶ Resolution limit
- ► Structure vs Noise



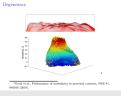
# Degeneracy



 $^4 Good$  et al., Performance of modularity in practical contexts, PRE 81, 046106 (2010).

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

-Degeneracy



- ► Maximizing the modularity can fail to resolve small sized modules
- ▶ Modular structures like cliques can be hidden in the larger groups of nodes with higher modularity score
- ▶ Peak of the modularity function may not coincide with divisions that identify such modular structures

$$Q = \frac{1}{2m} \sum_{i,j} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

Contribution of the group s,

$$Q_s = \frac{1}{2m} \sum_{i,j} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, s) \delta(c_j, s) = \frac{e_s}{m} - \left( \frac{d_s}{2m} \right)^2$$

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

Resolution limit Large-scale structure of complex networks (Part 2)

groups of nodes with higher modularity scor  $Q_s = \frac{1}{2m} \sum \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, s) \delta(c_j, s) = \frac{e_s}{m} - \left( \frac{d_s}{2m} \right)$ 

 $e_s$ : Number of links inside module s

-Resolution limit

 $d_s$ : sum of the degrees inside s

#### Resolution limit

The group s is a module whenever  $Q_s > 0 \Rightarrow \frac{e_s}{m} > \left(\frac{d_s}{2m}\right)^2$ 

Consider two modules  $s_1$  and  $s_2$  with  $e_{s_1s_2}$  edges between them The change in modularity if we merge these:

$$\triangle Q_{s_1 s_2} = \frac{e_{s_1 s_2}}{m} - 2\left(\frac{d_{s_1}}{2m}\right) \left(\frac{d_{s_2}}{2m}\right) > 0$$

whenever:

$$d_{e_1}d_{e_2}$$

 $e_{s_1 s_2} > \frac{d_{s_1} d_{s_2}}{2m} \to 0$ 

Thus, modules would be merged even when the number of links  $e_{s_1s_2}$  between them is small! <sup>5</sup>

-Resolution limit

Large-scale structure of complex networks (Part 2)

High wavelength light

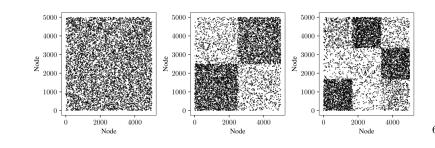
The group s is a module whenever  $O_* > 0 \Rightarrow \stackrel{\iota}{\Rightarrow} > (\stackrel{\iota}{\Rightarrow})^2$ 

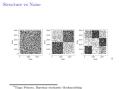
Resolution limit

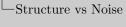
PNAS, (2006) 4日 → 4団 → 4 豆 → 4 豆 → 9 へ ○

<sup>&</sup>lt;sup>5</sup>Fortunato, Barthelemy, Resolution limit in community detection,

#### Structure vs Noise



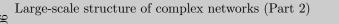




Large-scale structure of complex networks (Part 2)

#### Conclusions

- ► Community structure is a fundamental property of networks
- ▶ Community detection is an ill-defined problem
- ▶ (Too) many algorithms exist
- ▶ Community detection is still an open problem!







► Community detection is an ill-defined problem

Conclusions

- (Too) many almorithms exist
- $\blacktriangleright$  Community detection is still an open problem!

4 D > 4 D > 4 E > 4 E > E 990