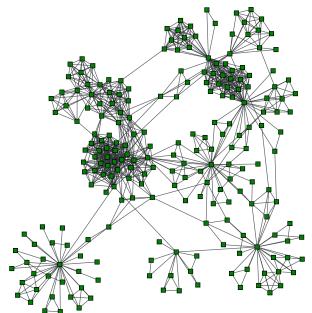
Snehal M. Shekatkar

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

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)

2018-01-01

-Community structure in networks

Community structure in networks

What are communities?

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

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
- 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

2018-01-01

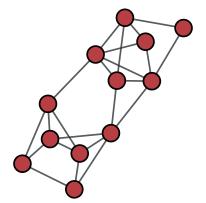
Community detection

Detecting communities is important!

- Communities are building blocks of network
- \blacktriangleright Communities allow us to see "the big pictur
- 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

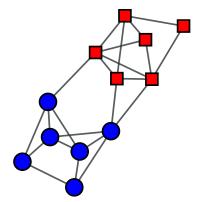
2018-01-01

Problem of dividing a graph in a given number of georape of given sizes such that the number of fluids between the groups (cut skep) is minimized.

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 is a given number of groups of green sizes such that we under of flash between the groups (and alley) and the state of t

Graph partitioning

-Graph partitioning

2018-01-01

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!

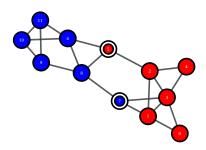
Large-scale structure of complex networks (Part 2)

- Cough with a vortices
- Partitioning is hard!

- Partitioning is hard!

- Partitioning is hard!

cut size = 4

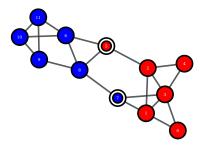


▶ Divide the vertices into two groups of the required sizes and calculate the cut size Large-scale structure of complex networks (Part 2)

└─Kernighan-Lin algorithm



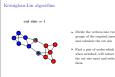
cut size = 4



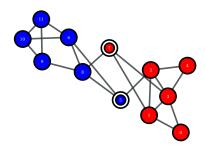
- ➤ Divide the vertices into two groups of the required sizes and calculate the cut size
- ► Find a pair of nodes which when switched, will reduce the cut size most and switch them

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

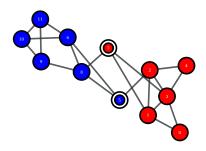
Large-scale structure of complex networks (Part 2)

└─Kernighan-Lin algorithm

2018-01-01



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

2018-01-01

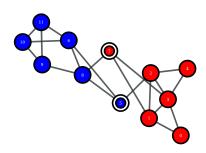


Kernighan-Lin algorithm

Find a pair of modes white when switched, will reduce the cut size most and switherm

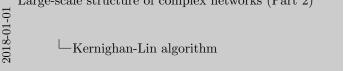
If no such pair exists, selethe pair which least increthe cut size

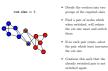
cut size = 2



- ▶ 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
- ► Continue this such that the already switched pair is not switched again

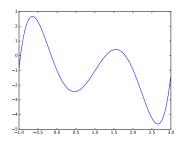
Large-scale structure of complex networks (Part 2)





Kernighan-Lin algorithm

· 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

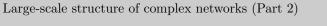
Kernighan-Lin algorithm

Continue till the cut size no

► Go through all the states and

Group sizes remain constant

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





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

Spectral partitioning



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

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

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

Large-scale structure of complex networks (Part 2)

Large-scale structure of complex networks (Part 2) $n = \frac{1}{2} \sum_{\substack{\text{Odds} \\ \text{Part of limited}}} \sum_{\substack{\text{The other limits} \\ \text{The other limits}}} \sum_{\substack{\text{The other$

Previous perturbating $R = \frac{1}{2} \sum_{j,m} A_{ij}$ Define $s_{ij} = \begin{cases} +1 & \text{if wetter it belongs to group 1} \\ -1 & \text{if wetter it belongs to group 2} \end{cases}$ Then $\frac{1}{2}(1-sx_{ij}) = \begin{cases} 1 & \text{if and } j \text{ are in different groups,} \\ 0 & \text{if and } j \text{ are in different groups,} \end{cases}$

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

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)

2018-01-0

└Spectral partitioning

Spectral partitioning
$$\begin{split} R &= \frac{1}{4} \sum_{ij} A_{ij} (1 - \kappa \nu_i) \end{split}$$
 First term, $\sum_{ij} A_{ij} &= \sum_{j} k_i v_j^2 - \sum_{ij} k_i k_{ij} \kappa \nu_j \\ R &= \frac{1}{4} \sum_{ij} (k_i l_{ij} - A_{ij}) \kappa \nu_j - \frac{1}{2} \sum_{ij} L_{ij} \kappa \nu_i \end{split}$ $R = \frac{1}{4} \sum_{ij} (k_i l_{ij} - A_{ij}) \kappa \nu_j - \frac{1}{4} \sum_{ij} L_{ij} \kappa \nu_i$

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

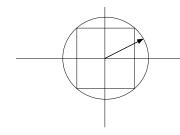
s is a column vector

L: structure, s: division

find s that minimizes R

Problem is hard, s takes only integer values

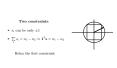
$$\triangleright$$
 s_i can be only ± 1



Relax the first constraint

Large-scale structure of complex networks (Part 2)

Relaxation method



Relaxation method

hypercube

continuous s, differentiate

Minimization with constraints \Rightarrow Lagrange multipliers

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

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

Spectral partitioning

-Spectral partitioning

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

$$\sum_j L_{jk} s_j - \lambda s_i + \mu$$

Minimization with constraints \Rightarrow Lagrange multipliers

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

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

-Spectral partitioning

-Spectral partitioning

Minimization with constraints \Rightarrow Lagrange multipliers

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

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

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

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

1 is an eigenvector of the Laplacian with eigenvalue 0

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

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

 ${\bf x}$ 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
$$\begin{split} \mathbf{x} &\text{ is an eigenvector of the Laplacian with eigenvalue λ} \\ &\text{Which eigenvector to choose?} \\ &\text{ \mathbf{x} cannot be the eigenvector } \mathbf{1} = \begin{bmatrix} 1\\ 1\\ 1 \end{bmatrix}, \\ \mathbf{1}^T\mathbf{x} &= \mathbf{1} \left(\mathbf{x} + \frac{\pi}{\Lambda}\mathbf{1}\right) = (n_1 - n_2) + \frac{\pi}{\Lambda}\mathbf{x} = 0. \end{split}$$

x is orthogonal to 1

x is eigenvector but not 1

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? $R = \frac{1}{4}\mathbf{s}^{T}\mathbf{L}\mathbf{s} = \frac{1}{4}\mathbf{x}^{T}\mathbf{x} = \frac{n_{1}n_{2}}{2}\lambda$

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

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

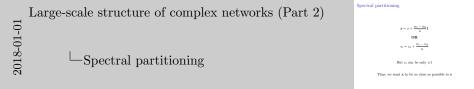
$$\mathbf{s} = x + \frac{n_1 - n_2}{n} \mathbf{1}$$

\mathbf{OR}

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

But s_i can be only ± 1

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



Maximize:

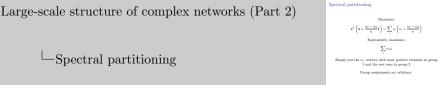
$$\mathbf{s}^T \left(\mathbf{x} + \frac{n_1 - n_2}{n} \mathbf{1} \right) = \sum_i s_i \left(x_i + \frac{n_1 - n_2}{n} \right)$$

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

- \triangleright Calculate \mathbf{v}_2 of the Laplacian
- \triangleright Put vertices corresponding to largest n_1 elements in group 1 and others in group 2. Calculate the cut size
- \triangleright 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

- Calculate v₂ 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
- · Choose the division with the smallest cut size among the

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!

2018-01-01

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

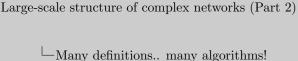
ill-defined

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

4 D > 4 A > 4 B > 4 B > B 9 Q Q

Many definitions.. many algorithms!

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



2018-01-01

Many definitions.. many algorithms!

► 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

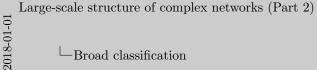
▶ Divisive algorithms:

- ► Girvan-Newman algorithm
- ▶ Radichhi algorithm

► Assignment algorithms:

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

◆ロト ◆部ト ◆恵ト ◆恵ト ・恵 ・ 釣り(で)



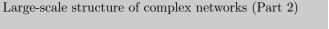


► Agglomerative algorithms:

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

"The" simplest community detection problem

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





"The" simplest community detection problem

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

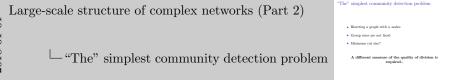
Empty group

2018-01-01

"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

Large-scale structure of complex networks (Part 2)

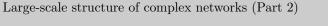
Quantification of community structure $\bullet \mbox{ Fewer than expected edges between the groups }$

2018-01-0

—Quantification of community structure

few edges = expected edges = not a good division

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



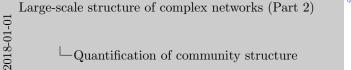
└─Quantification of community structure

Quantification of community structure

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

Remember assortativity

- ▶ 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

- · Assortativity mixing and modularity

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)

2018-01-01

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 modul
- Look for divisions with h

Heuristics are needed

- ► 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)

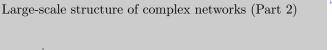
2018-01-01

Quantification of community structure

Quantification of community structure

- ► Fewer than expected edges between the grouns
- Equivalently, more than expected edges inside the g
- Assortativity mixing and modulari
- Look for divisions with high me
- ► Modularity maximization is hard

▶ Start with a random division of the nodes



Kernighan-Lin-Newman algorithm

-Kernighan-Lin-Newman algorithm

Variation of KL algorithm

Sizes of the groups are not fixed

- ▶ 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)

2018-01-01

-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

ith a random division of the nodes

 ${}^{\blacktriangleright}$ Change in modularity for shifting each vertex to the other group

Variation of KL algorithm

Sizes of the groups are not fixed

- ▶ 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)

2018-01-01

-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

- ▶ 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)

2018-01-01

-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 best decrease in the modularity.

an Im 110 man angorrami

Variation of KL algorithm

Sizes of the groups are not fixed

- ▶ 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)

2018-01-0

-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

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

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

◆ロト ◆個ト ◆意ト ◆意ト 意 めなべ

- ▶ 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
- \blacktriangleright 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)

-Kernighan-Lin-Newman algorithm

Kernighan-Lin-Newman algorithm

- ${}^{\bullet}$ 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
- least decrease in the modularity
- repeat so that the vertex once moved is not moved
- ▶ Select a state with the highest modularity

Variation of KL algorithm

Sizes of the groups are not fixed

No swapping

◆ロト ◆母ト ◆意ト 意 めの○

- ► 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)

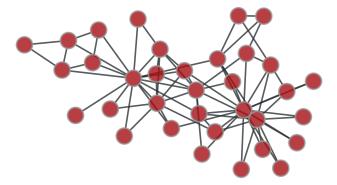
-Kernighan-Lin-Newman algorithm

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

Variation of KL algorithm

Sizes of the groups are not fixed



Large-scale structure of complex networks (Part 2)



Does somebody know this network?

Zachry karate club network

10-01-01

Large-scale structure of complex networks (Part 2)

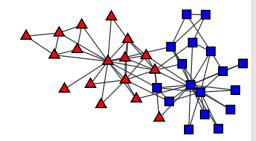


Zachry karate club network

2018-01-01

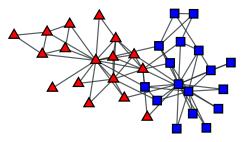
Zachry karate club network





Application to Zachry karate club

Actual division



Large-scale structure of complex networks (Part 2)

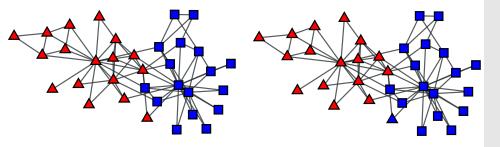
LApplication to Zachry karate club



Application to Zachry karate club

Actual division

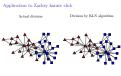
Division by KLN algorithm



Large-scale structure of complex networks (Part 2)

2018-01-01

LApplication 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

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

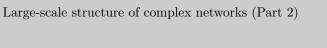
Spectral modularity maximization

spectral partitioning: cut size

analogous algorithm exists

Spectral modularity maximization

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





2018-01

-Spectral modularity maximization

Spectral modularity maximization

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

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

Spectral modularity maximization Large-scale structure of complex networks (Part 2) -Spectral modularity maximization

◆ロト ◆部ト ◆恵ト ◆恵ト ・恵 ・ 釣り(で)

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

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

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

Large-scale structure of complex networks (Part 2)

-Spectral modularity maximization

 $s_i = \begin{cases} +1 & \text{if we$ $tex } 1 \text{ belong to group 1} \\ -1 & \text{if we$ $tex } 1 \text{ belong to group 2} \end{cases}$ $\frac{1}{2}(1+s,s_d) = \begin{cases} -1 & \text{if } s \text{ and } J \text{ belong to the same group} \\ 0 & \text{Otherwise} \end{cases}$ $B = \frac{1}{2m} \sum_{ij} R_{ij}(s_i,s_j) = \frac{1}{4m} \sum_{ij} R_{ij}(+s_{s^2}) = \frac{1}{4m} \sum_{ij} R_{ij}s_{s^2}$ $Q = \frac{1}{4m} s^2 \text{Ba}.$

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

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

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

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

Spectral modularity maximization Large-scale structure of complex networks (Part 2) Numbers of elements with values +1 and -1 are not fixed Only constraint: s^Ts = ∑s_i² = n -Spectral modularity maximization

s is eigenvector of modularity matrix

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 **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 $u_{1i}!$

Spectral modularity maximization Large-scale structure of complex networks (Part 2) -Spectral modularity maximization

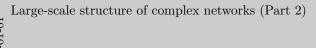
 $Q = \frac{1}{1} \beta \mathbf{s}^T \mathbf{B} \mathbf{s} - \frac{1}{1} \beta \mathbf{s}^T \mathbf{s} - \frac{n}{1} \beta$ gest eigenvalue of the modularity matrix $\mathbf{s}^T \mathbf{u}_1 = \sum s_i u_{1i}$ Maximum is achieved when each term is non-negative \Rightarrow Us

-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

Newman-Girvan algorithm

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



Newman-Girvan algorithm

Look for edges between the communities

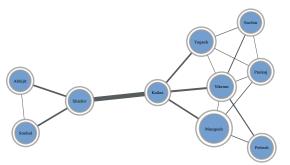
► Edge betweenness

Let's have a look at the edge betweenness

└─Newman-Girvan algorithm

Edge betweenness

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



4□ > 4□ > 4□ > 4□ > 4□ > 3

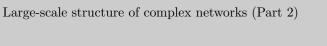
Large-scale structure of complex networks (Part 2)

- Path between two modes
- State tot path between two modes
- Number of shortest path that got through a given alogs

- 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

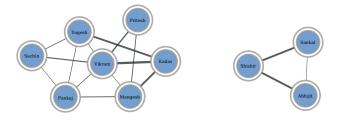
- ► Calculate betweenness for all edges
- ► Remove the edge with the highest betw
- ► Recalculate betweenness for all edges
- Repeat

4□ > 4□ > 4 = > 4 = > = 900



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

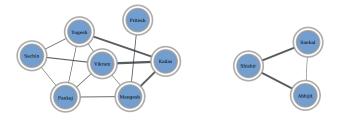




Large-scale structure of complex networks (Part 2)



Girvan-Newman algorithm

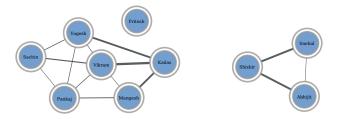


Large-scale structure of complex networks (Part 2)



Girvan-Newman algorithm



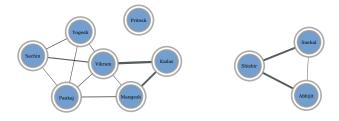


Large-scale structure of complex networks (Part 2)





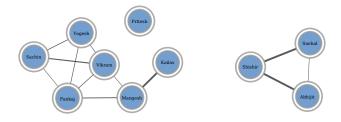
Girvan-Newman algorithm



Large-scale structure of complex networks (Part 2)



Girvan-Newman algorithm



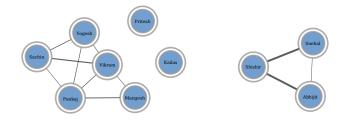
Large-scale structure of complex networks (Part 2)

└─Girvan-Newman algorithm





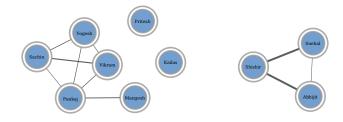




Large-scale structure of complex networks (Part 2)





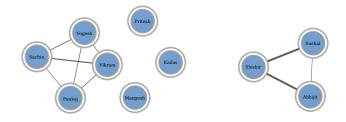


Large-scale structure of complex networks (Part 2)

Girvan-Newman algorithm



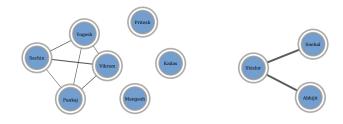




Large-scale structure of complex networks (Part 2)

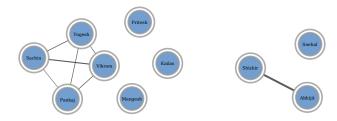


Girvan-Newman algorithm



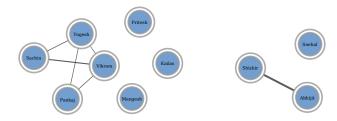
Large-scale structure of complex networks (Part 2)

Girvan-Newman algorithm



Large-scale structure of complex networks (Part 2)

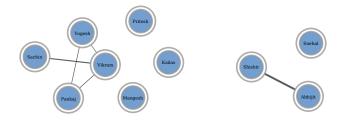
Girvan-Newman algorithm



Large-scale structure of complex networks (Part 2)



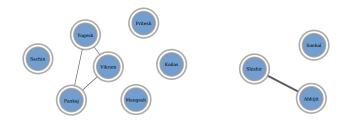
Girvan-Newman algorithm



Large-scale structure of complex networks (Part 2)



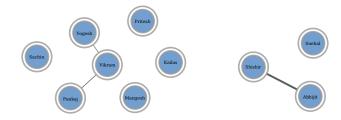
Girvan-Newman algorithm



Large-scale structure of complex networks (Part 2)

Girvan-Newman algorithm





Large-scale structure of complex networks (Part 2)

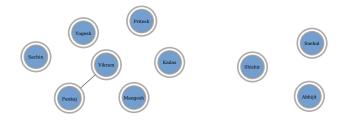
Girvan-Newman algorithm

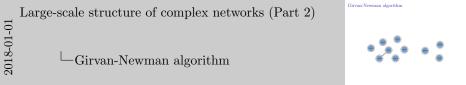


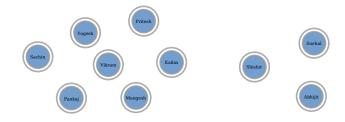


Girvan-Newman algorithm Large-scale structure of complex networks (Part 2) Girvan-Newman algorithm









Large-scale structure of complex networks (Part 2)

Girvan-Newman algorithm