Large-scale structure of complex networks (Part 2)

Snehal M. Shekatkar

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

2017-12-29

Large-scale structure of complex networks (Part 2)

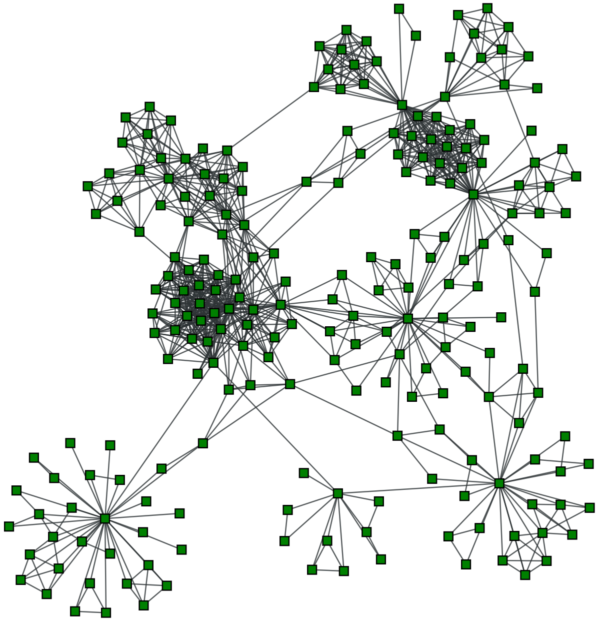
Large-scale structure of complex networks (Part 2)

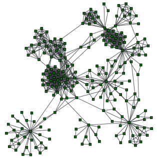
Snehal M. Shekatkar

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

Hello

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

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

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 network**:
  - Proteins with similar roles in protein interaction networks
  - Chemicals together taking part in chemical reactions in metabolic networks
  - Communities in neuronal 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)
Biological network: 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

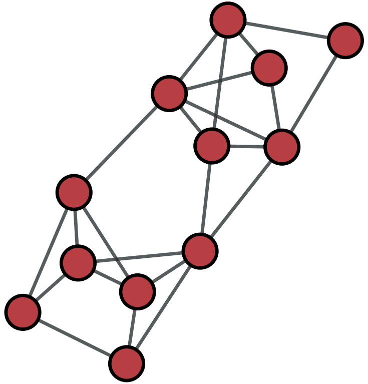
Community detection

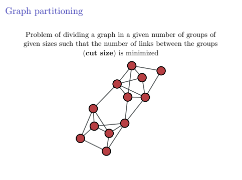
Detecting communities is important!

- 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

Graph partitioning

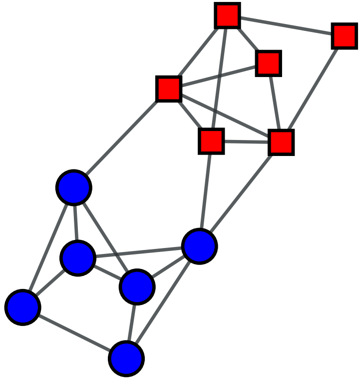
Problem of dividing a graph in a given number of groups of given sizes such that the number of links between the groups (**cut size**) is minimized

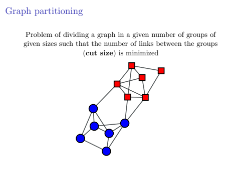




Graph partitioning

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





Partitioning is hard!

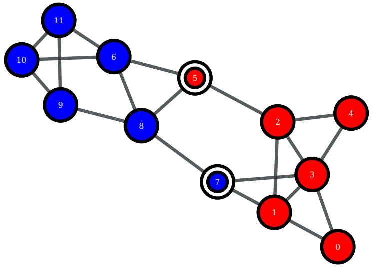
- Graph with n vertices
- Find two groups with sizes n_1 and n_2 such that the cut size is minimum
- Number of ways: $\frac{n!}{n_1!n_2!} \approx \frac{2^{n+1}}{\sqrt{n}}$

Heuristics are needed!

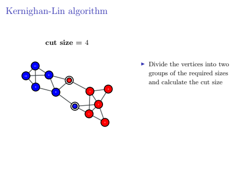
Partitioning is hard!
- Graph with n vertices
- Find two groups with sizes n_1 and n_2 such that the cut size is minimum
- Number of ways: $\frac{n!}{n_1!n_2!} \approx \frac{2^{n+1}}{\sqrt{n}}$
Heuristics are needed!

Kernighan-Lin algorithm

cut size $= 4$

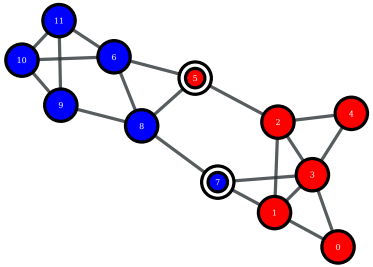


- Divide the vertices into two groups of the required sizes and calculate the cut size

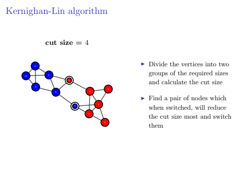


Kernighan-Lin algorithm

cut size $= 4$

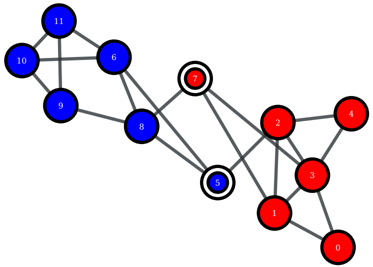


- Divide the vertices into two groups of the required sizes and calculate the cut size
- Find a pair of nodes which when switched, will reduce the cut size most and switch them

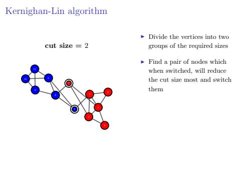


Kernighan-Lin algorithm

cut size = 2

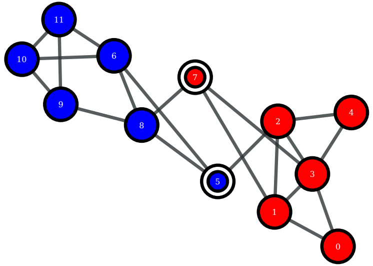


- Divide the vertices into two groups of the required sizes
- Find a pair of nodes which when switched, will reduce the cut size most and switch them



Kernighan-Lin algorithm

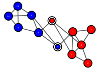
cut size = 2



- 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
- If no such pair exists, select the pair which least increases the cut size

Kernighan-Lin algorithm

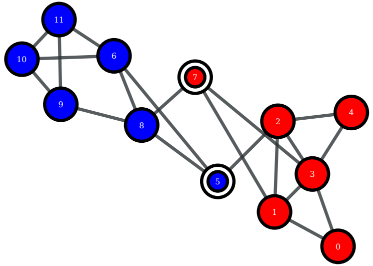
cut size = 2



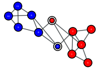
- Divide the vertices into two groups of the required sizes
- Find a pair of nodes which when switched, will reduce the cut size most and switch them
- If no such pair exists, select the pair which least increases the cut size

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
- 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

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
- 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

Kernighan-Lin algorithm

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

Kernighan-Lin algorithm

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

Group sizes remain constant

Spectral paritioning

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

Spectral paritioning

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

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

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

Many definitions.. many algorithms!

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

Many definitions.. many algorithms!
- Girvan-Newman algorithm
- Kernighan-Lin-Newman algorithm
- Spectral decomposition
- Clique-percolation
- Radom walk methods
- 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

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

“The” simplest community detection problem

- 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

"The" simplest community detection problem

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

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

"The" simplest community detection problem
- 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

Quantification of community structure

- Fewer than expected edges between the groups

Quantification of community structure
- Fewer than expected edges between the groups

few edges = expected edges = not a good division

Quantification of community structure

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

Quantification of community structure

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

Remember assortativity

Quantification of community structure

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

Quantification of community structure
- Fewer than expected edges between the groups
- Equivalently, more than expected edges inside the groups
- Assortativity mixing and modularity

Divide network using modularity

Quantification of community structure

- 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

Quantification of community structure

- 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

Heuristics are needed

Quantification of community structure

- 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

Quantification of community structure

- 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

Newman-Girvan algorithm

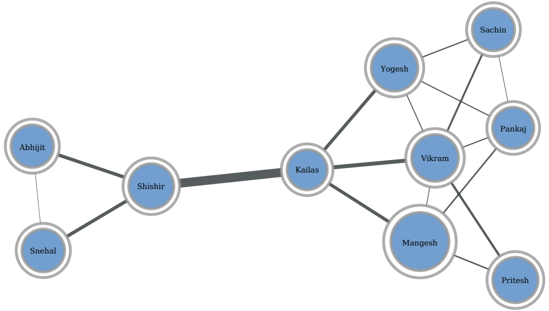
- Look for edges between the communities
- Edge betweenness

Newman-Girvan algorithm
- Look for edges between the communities
- Edge betweenness

Let's have a look at the edge betweenness

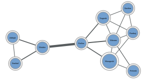
Edge betweenness

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



Edge betweenness

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



The algorithm

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

The algorithm

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