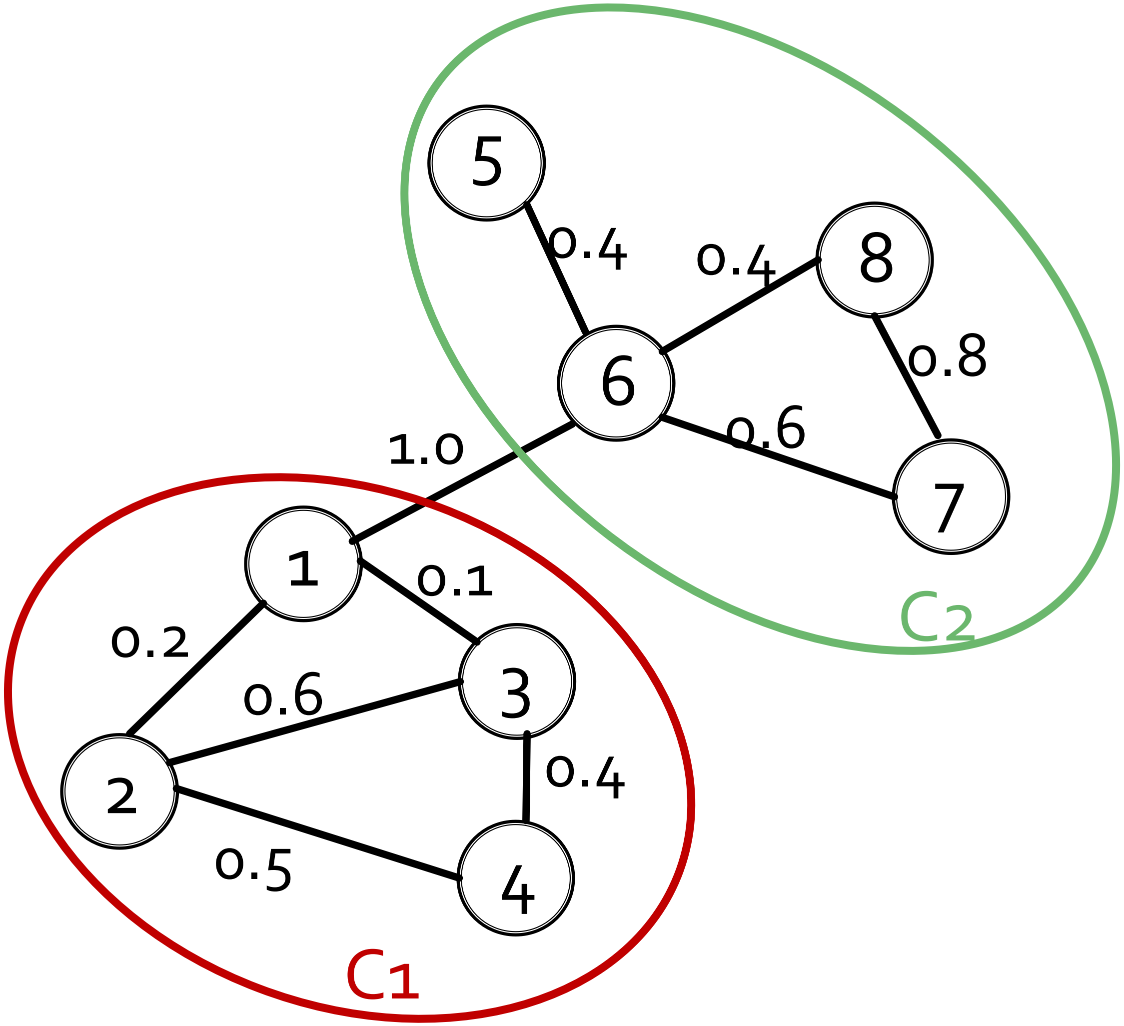
Graph community detection Tutorial

Lin Guosheng
School of Computer Science and Engineering
Nanyang Technological University

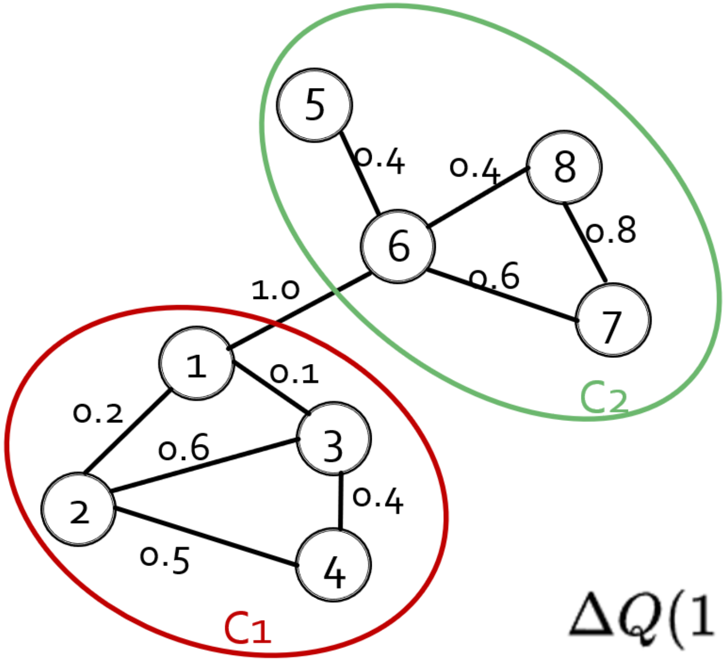
Question 1

Q: A graph is given below.
The current community assignment is:
C1={1, 2, 3, 4},
C2={5,6,7,8}.

Process node #1 to update the communities using Louvain Algorithm.


5
0.4
0.4
8
0.8
6
0.6
1.0
7
C2
1
0.1
0.2
0.6
3
0.4
2
0.5
4
C1

Solution



Process node 1:

1) There is only one neighbouring communities of node 1:
Community C2

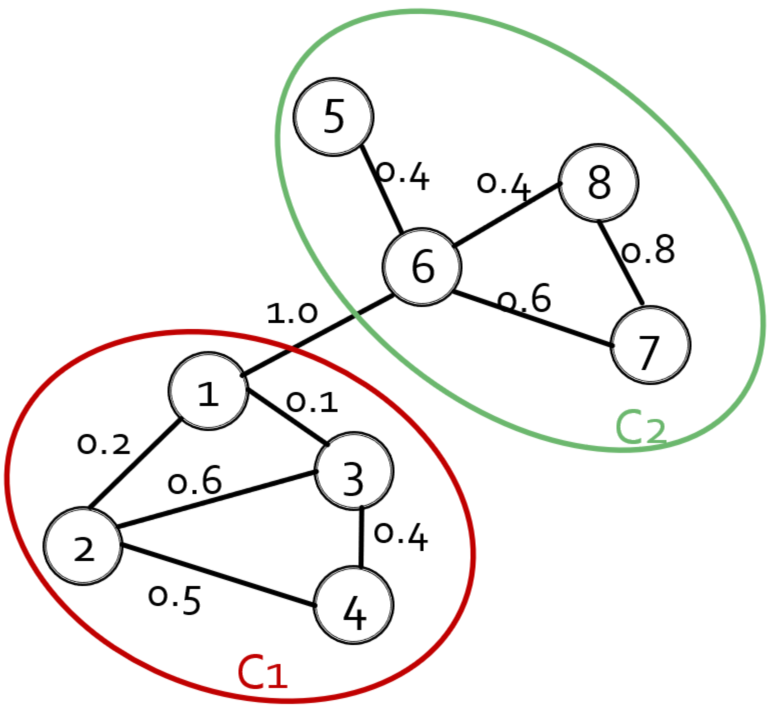
2) Only one movement:
We need to calculate the modularity gain (movement score) of moving node 1 to the neighbouring communities :

$$\Delta Q(1 \rightarrow \{5,6,7,8\}) = Q_{after} - Q_{before}$$

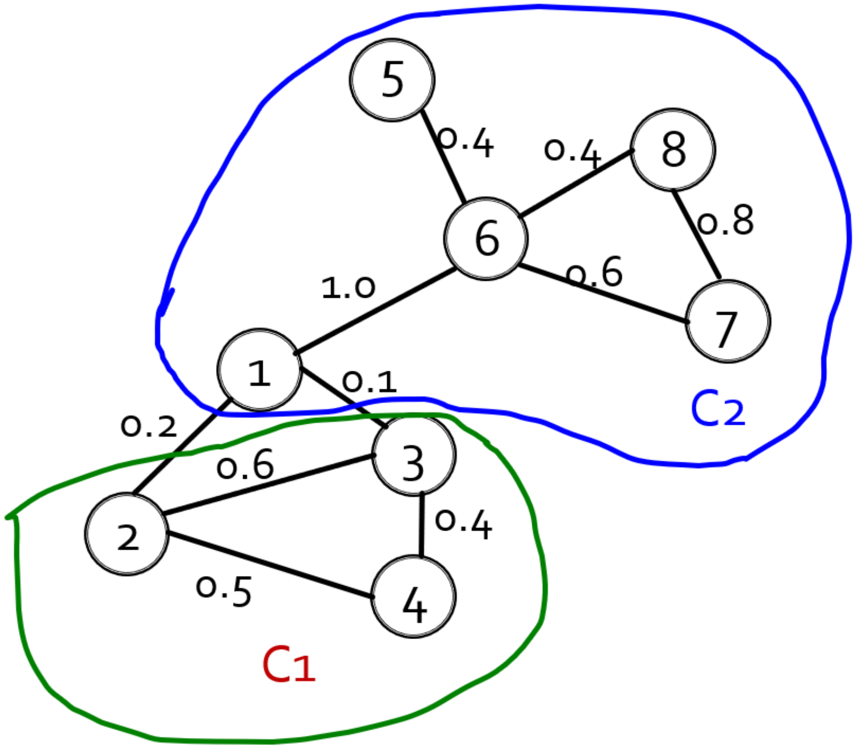
Solution

Process node 1:

$$\Delta Q(1 \rightarrow \{5,6,7,8\}) = Q_{after} - Q_{before}$$
$$= [Q(\{1,5,6,7,8\}) + Q(\{2,3,4\})] - [Q(\{5,6,7,8\}) + Q(\{1,2,3,4\})]$$



Q_{before}



Q_{after}

Solution

$$\Delta Q(1 \to \{5,6,7,8\}) \quad = \quad Q_{after} - Q_{before}$$

$$= \quad [Q(\{1,5,6,7,8\}) + Q(\{2,3,4\})] - [Q(\{5,6,7,8\}) + Q(\{1,2,3,4\})]$$

$$m = 0.2 + 0.6 + 0.5 + 0.1 + 0.4 + 1.0 + 0.4 + 0.4 + 0.6 + 0.8 = 5$$

$$Q(\{1,2,3,4\})$$

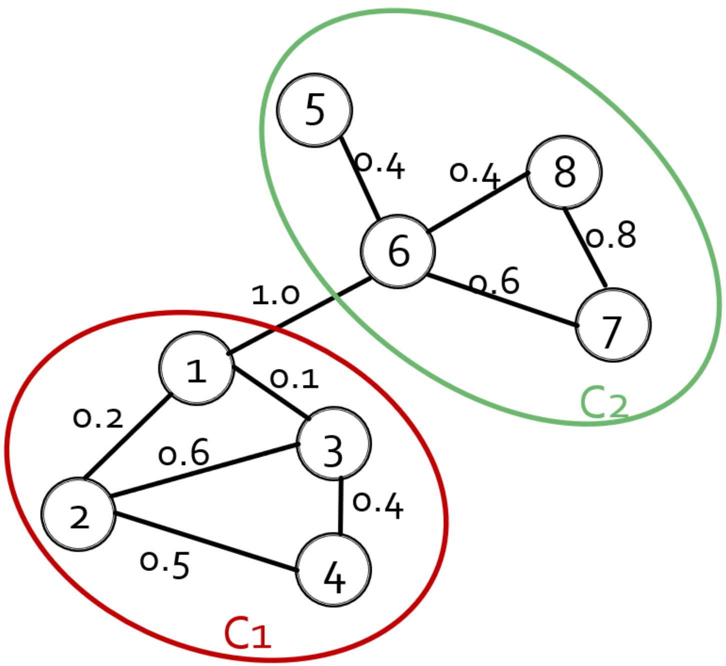
$$= \quad \frac{\Sigma_{in}}{2m} - (\frac{\Sigma_{tot}}{2m})^2$$

$$= \quad \frac{2\times(e_{12}+e_{13}+e_{23}+e_{24}+e_{34})}{2m} - (\frac{k_1+k_2+k_3+k_4}{2m})^2$$

$$= \quad \frac{2\times(0.2+0.1+0.6+0.5+0.4)}{10} - (\frac{1.3+1.3+1.1+0.9}{10})^2$$

$$= \quad 0.36 - 0.21$$

$$= \quad 0.15$$

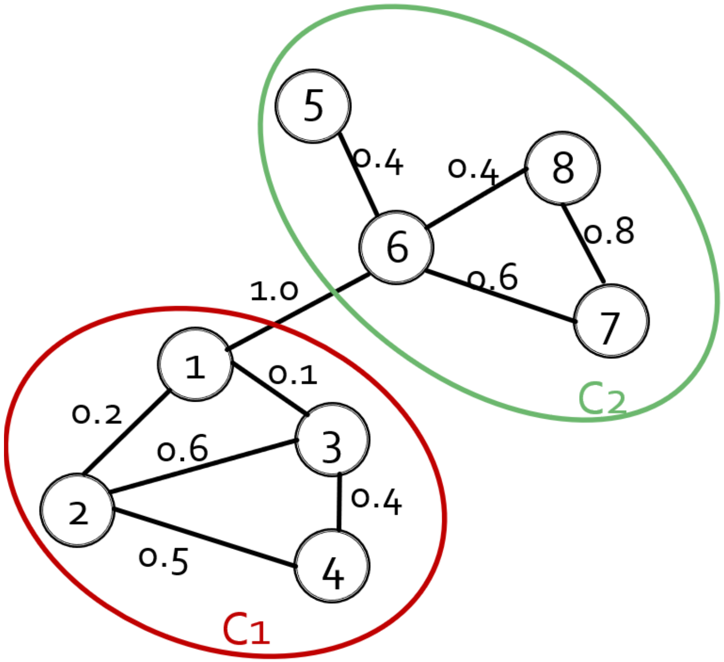


$$\Delta Q(1 \rightarrow \{5,6,7,8\}) \quad = \quad Q_{after} - Q_{before}$$

$$= \quad [Q(\{1,5,6,7,8\}) + Q(\{2,3,4\})] - [Q(\{5,6,7,8\}) + Q(\{1,2,3,4\})]$$

$$Q(\{5,6,7,8\})$$

$$
\begin{aligned}
&= \quad \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 \\
&= \quad \frac{2\times(e_{56}+e_{67}+e_{68}+e_{78})}{2m} - \left(\frac{k_5+k_6+k_7+k_8}{2m}\right)^2 \\
&= \quad \frac{2\times(0.4+0.6+0.4+0.8)}{10} - \left(\frac{0.4+2.4+1.4+1.2}{10}\right)^2 \\
&= \quad 0.44 - 0.29 \\
&= \quad 0.15
\end{aligned}
$$

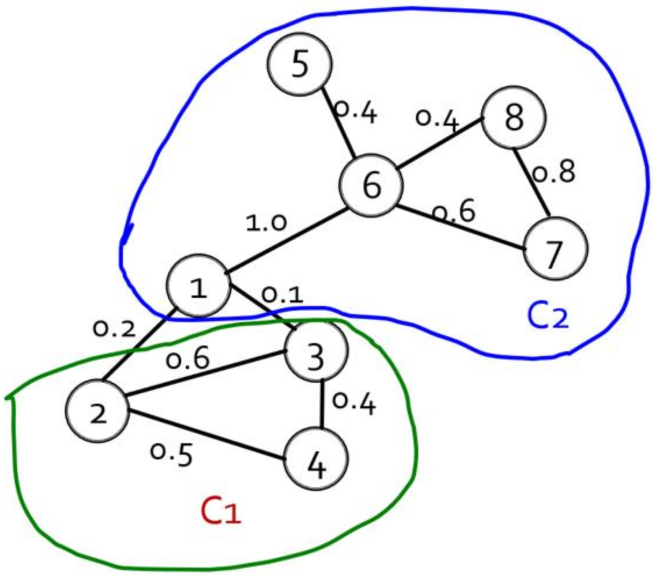


$$\Delta Q(1 \rightarrow \{5,6,7,8\}) \quad = \quad Q_{after} - Q_{before}$$

$$= \quad [Q(\{1,5,6,7,8\}) + Q(\{2,3,4\})] - [Q(\{5,6,7,8\}) + Q(\{1,2,3,4\})]$$

$$Q(\{2,3,4\})$$

$$\begin{aligned}
&= \quad \frac{\Sigma_{in}}{2m} - (\frac{\Sigma_{tot}}{2m})^2 \\
&= \quad \frac{2\times(e_{23}+e_{24}+e_{34})}{2m} - (\frac{k_2+k_3+k_4}{2m})^2 \\
&= \quad \frac{2\times(0.6+0.5+0.4)}{10} - (\frac{1.3+1.1+0.9}{10})^2 \\
&= \quad 0.3 - 0.11 \\
&= \quad 0.19
\end{aligned}$$



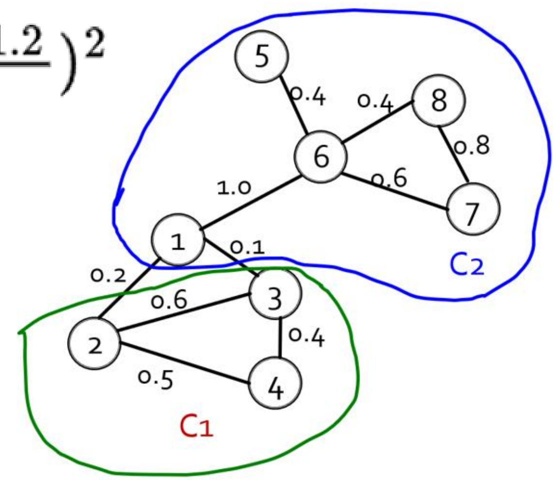
Q_{after}

$$\Delta Q(1 \rightarrow \{5,6,7,8\}) \quad = \quad Q_{after} - Q_{before}$$

$$= \quad \left[Q(\{1,5,6,7,8\}) + Q(\{2,3,4\})\right] - \left[Q(\{5,6,7,8\}) + Q(\{1,2,3,4\})\right]$$

$$Q(\{1,5,6,7,8\})$$

$$\begin{aligned}
&= \quad \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 \\
&= \quad \frac{2\times(e_{16}+e_{56}+e_{67}+e_{68}+e_{78})}{2m} - \left(\frac{k_1+k_5+k_6+k_7+k_8}{2m}\right)^2 \\
&= \quad \frac{2\times(1.0+0.4+0.6+0.4+0.8)}{10} - \left(\frac{1.3+0.4+2.4+1.4+1.2}{10}\right)^2 \\
&= \quad 0.64 - 0.45 \\
&= \quad 0.19
\end{aligned}$$



Q_{after}

$$\Delta Q(1 \rightarrow \{5,6,7,8\}) \quad = \quad Q_{after} - Q_{before}$$

$$= \quad [Q(\{1,5,6,7,8\}) + Q(\{2,3,4\})] - [Q(\{5,6,7,8\}) + Q(\{1,2,3,4\})]$$

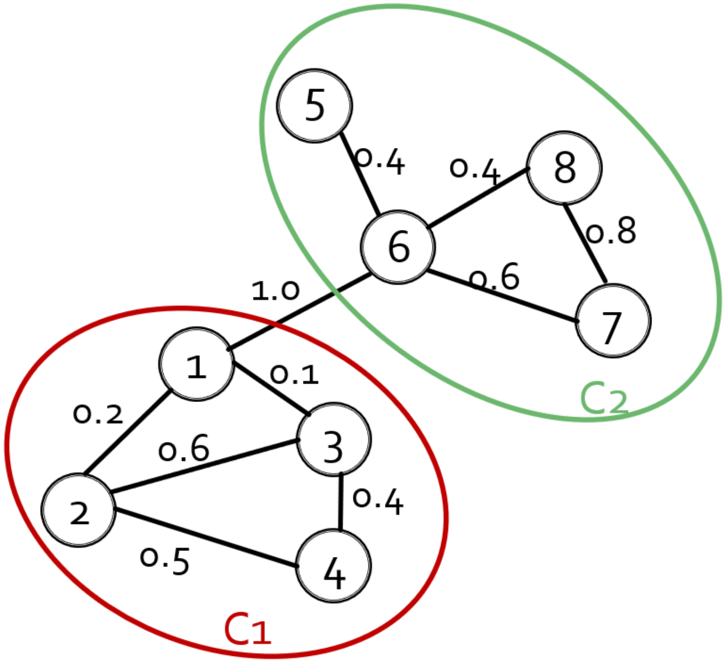
$$= \quad (0.19 + 0.19) - (0.15 + 0.15)$$

$$= \quad 0.08 > 0$$

There is only 1 neighbouring community for node 1.

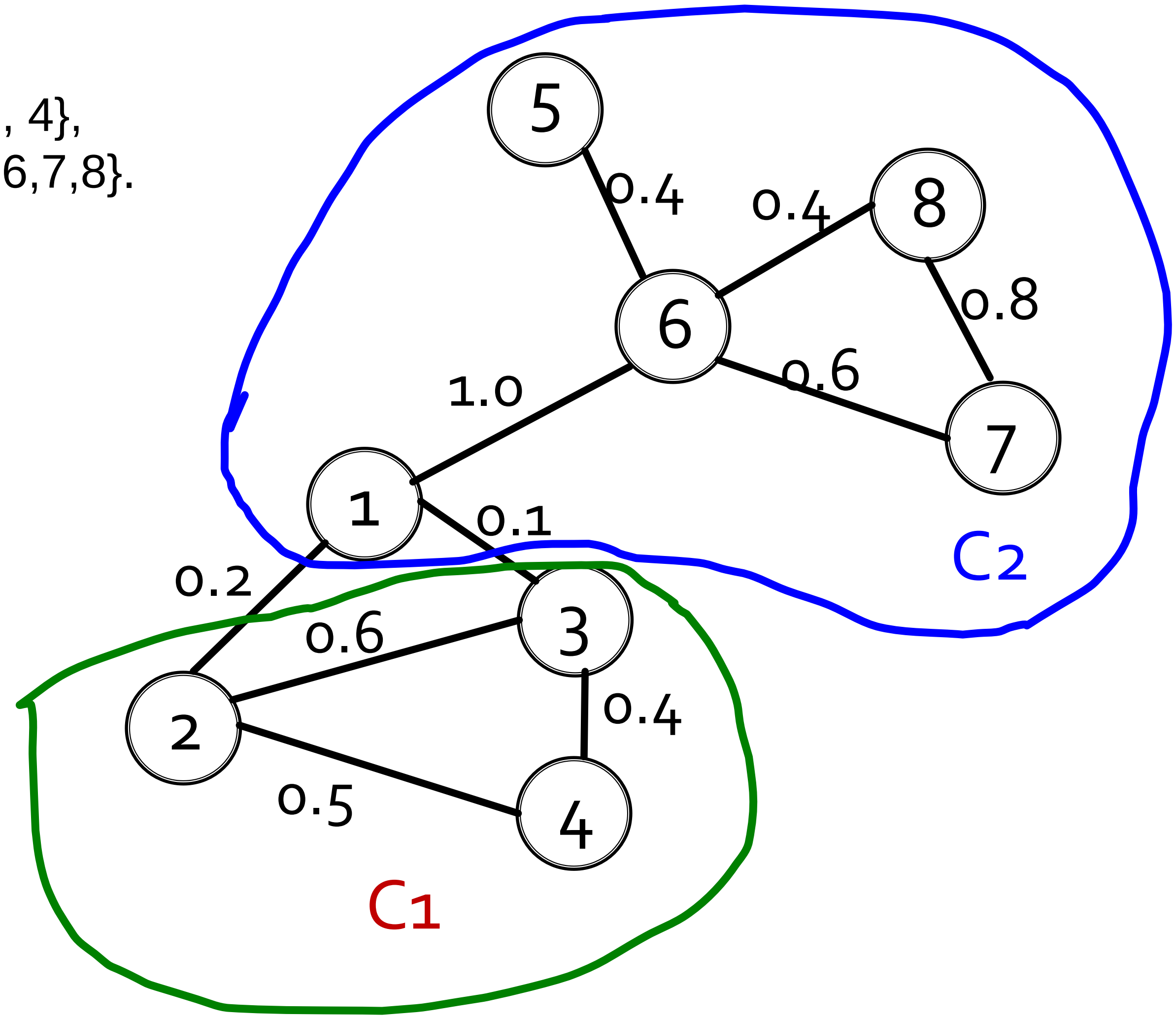
Moving node 1 to community C2 gives the largest positive modularity gain.

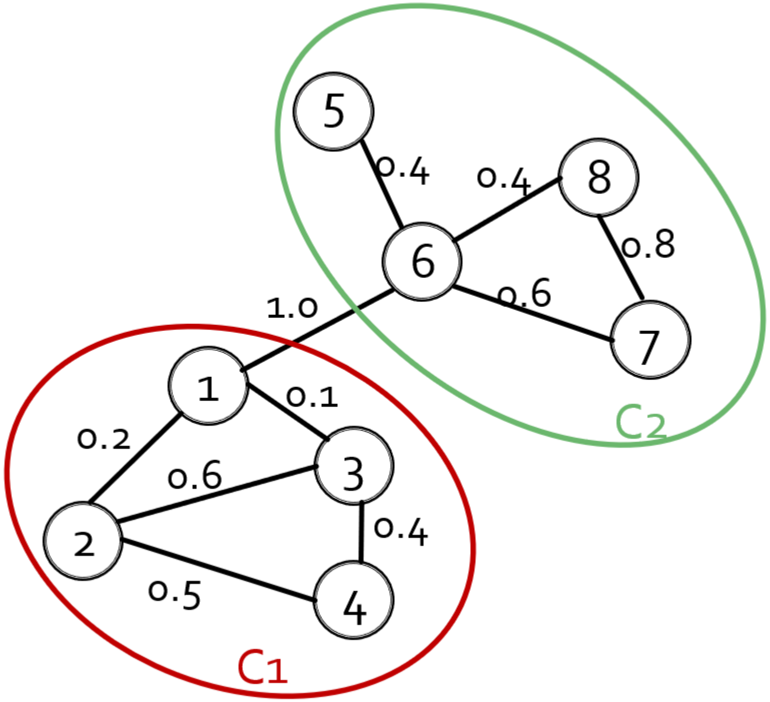
We move node 1 to community C2 to update the community assignment.



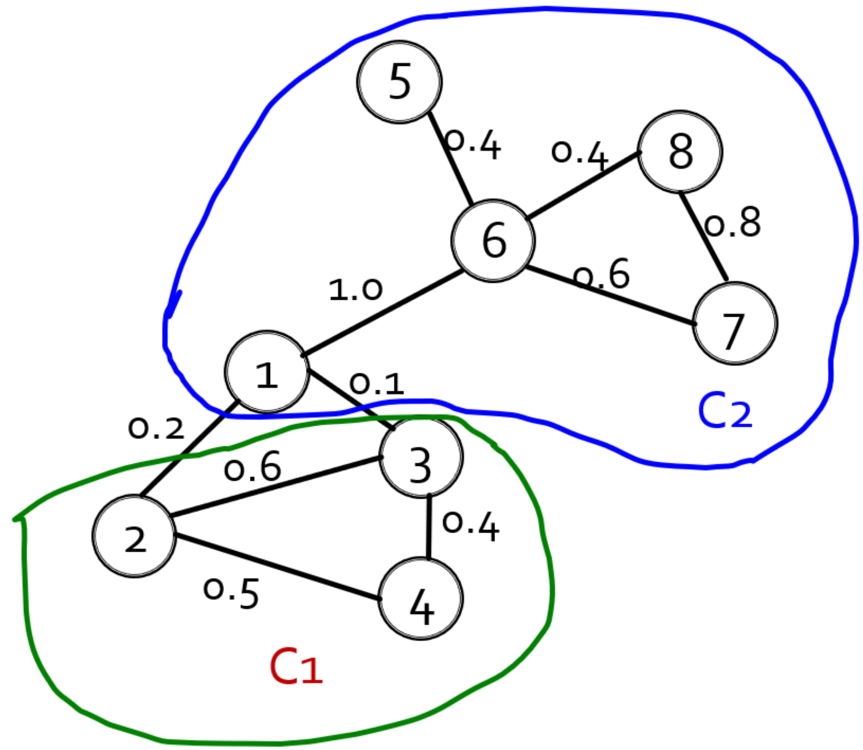
update the community assignment

C1={2, 3, 4},
C2={1,5,6,7,8}.





Before processing node 1



After processing node 1

Modularity score (Review)

Use this equation to calculate the modularity score for a community:

$$Q(C) = \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2$$

1. Modularity score: measure the quality of a community
2. We aim to maximize the total Modularity score of all communities.

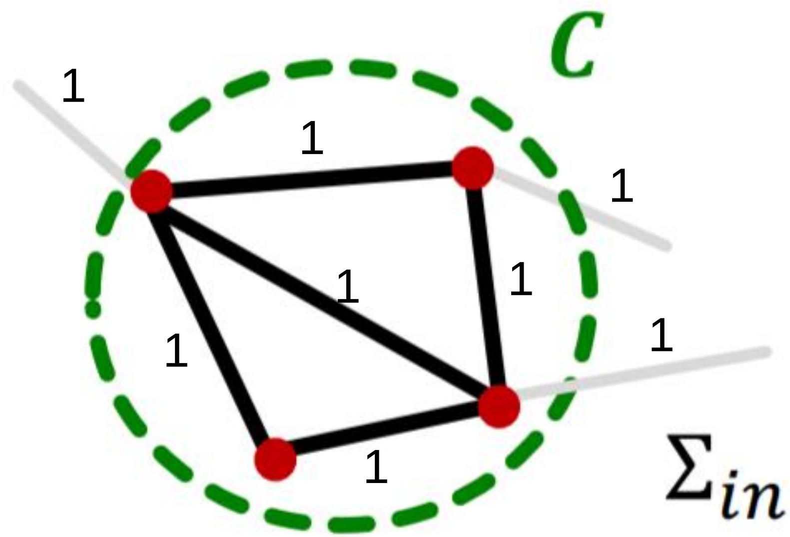
$\boldsymbol{\Sigma_{in}} \equiv \sum_{i,j \in C} A_{ij}$: Sum of the weights of internal directed edges in the community C.
(high value indicates strong internal connections)

$\boldsymbol{\Sigma_{tot}} \equiv \sum_{i \in C} k_i$: Sum of the weighted degrees of all nodes in the community C

$\boldsymbol{m}$: the sum of all edge weights in the undirected graph

For un-weighted graph: the weight for each edge is 1

Σ_{in}:



$$\Sigma_{in} = 10$$

The index i and j indicate the nodes in the community C

$\Sigma_{in} \equiv \sum_{i,j \in C} A_{ij}$: Sum of the weights of all internal directed edges in the community C

A_{ij} is the edge weight for the edge connecting the nodes (i, j)

In the undirected graph, the edge weight: $A_{i,j} = A_{j,i}$

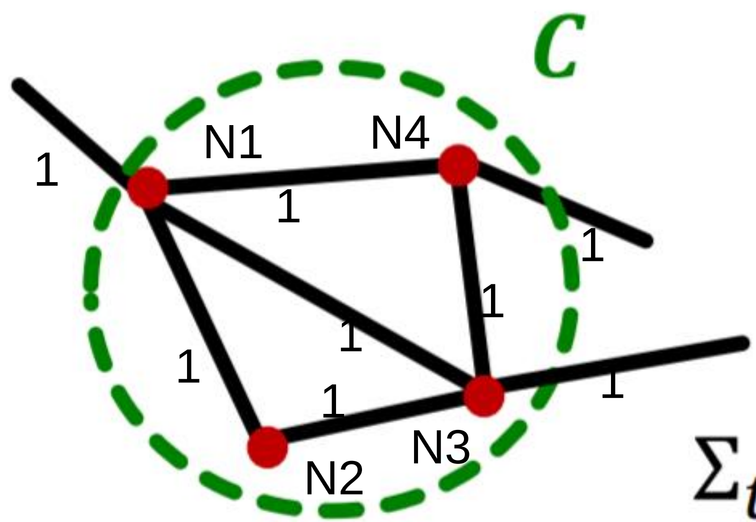
The internal edges will be double counted in the summation

For the example here, the internal edges will be double counted in the summation:

$$(1+1+1+1+1) \times 2 = 10$$

For un-weighted graph: the weight for each edge is 1

Σ_{tot}:



$$\Sigma_{tot} = 13$$

$\Sigma_{tot} \equiv \sum_{i \in C} k_i$: Sum of the weighted degrees of all nodes in the community C.
Here k_i is the weighted degree of node i.

Sum of the weighted degrees of all nodes in C:

4 + 2 + 4 + 3 = 13

(for node N1, N2, N3, N4, respectively)