SOCIAL MEDIA ANALYTICS INFS7450
Tutorial 8

Xia Xin, Xiangguo Sun

School of ITEE

The University of Queensland

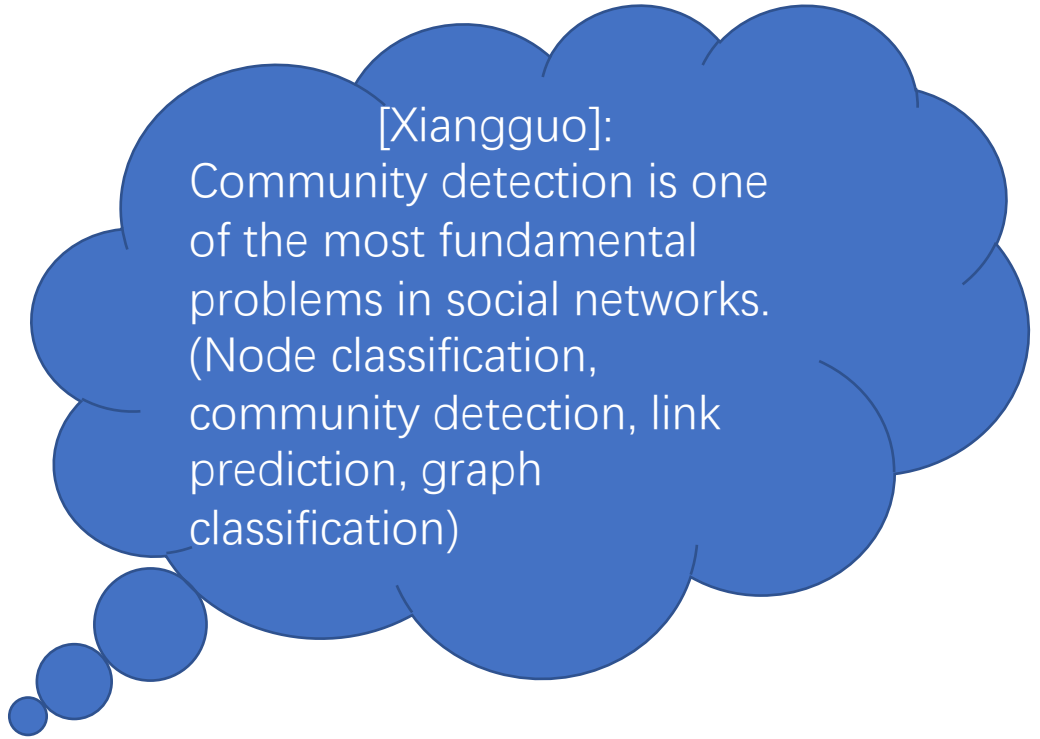
Social Community

[real-world] community

A group of individuals with common *economic*, *social*, or *political* interests or characteristics, often living in *relative proximity*.

What is Community Analysis?

- **Community detection**
  - Discovering implicit communities
- **Community evaluation**
  - Evaluating Detected Communities



What is community detection?

- The process of finding clusters of nodes (“*communities*”)
  - With **Strong** internal connections and
  - **Weak** connections between different communities

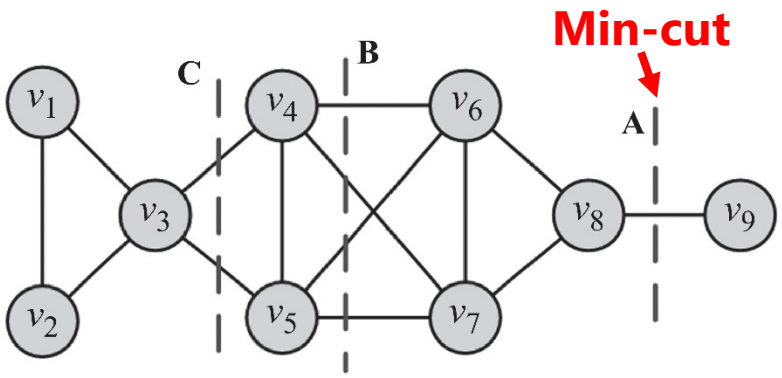
- Ideal decomposition of a large graph
  - Completely disjoint communities
  - There are no interactions between different communities.

- In practice,
  - find community partitions that are maximally decoupled.

Balanced Communities

[Xiangguo]:
If you are familiar with our tutorial week 2, you will also be familiar with this concept.

- Community detection can be viewed as *graph clustering*
- **Graph clustering:** we cut the graph into several partitions and assume these partitions represent communities
- **Cut**: partitioning (*cut*) of the graph into two (or more) sets (*cutsets*)
  - **The size of the cut** is the number of edges that are being cut
- **Minimum cut (min-cut) problem**: find a graph partition such that the number of edges between the two sets is minimized



Min-cuts can be computed efficiently using the max-flow

Min-cut often returns an imbalanced partition, with one set being a singleton

Ratio Cut and Normalized Cut

- To mitigate the min-cut problem we can change the objective function to consider community size

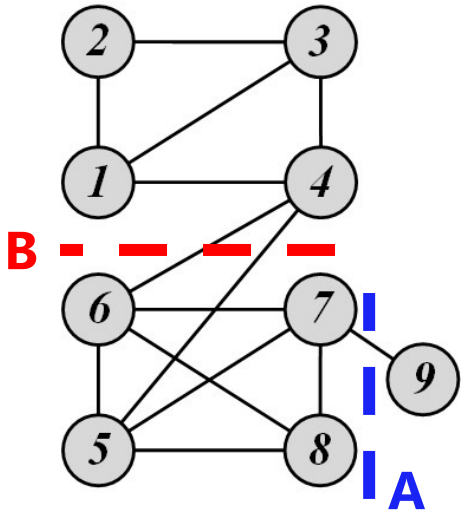
$$\text{Ratio Cut}(P) = \frac{1}{k}\sum_{i=1}^{k}\frac{\text{cut}(P_i,\bar{P}_i)}{|P_i|}$$

$$\text{Normalized Cut}(P) = \frac{1}{k}\sum_{i=1}^{k}\frac{\text{cut}(P_i,\bar{P}_i)}{\text{vol}(P_i)}$$

- $\bar{P}_i = V - P_i$ is the complement cut set
- $cut(P_i, \bar{P}_i)$ is the size of the cut
- $vol(P_i) = \sum_{v \in P_i} d_v$

Ratio Cut & Normalized Cut: Example

[Xiangguo]:
This might be in your final exam



For Cut A

$$\text{Ratio Cut}(\{1,2,3,4,5,6,7,8\},\{9\}) = \frac{1}{2}(\frac{1}{1}+\frac{1}{8}) = 9/16 = 0.56$$

$$\text{Normalized Cut}(\{1,2,3,4,5,6,7,8\},\{9\}) = \frac{1}{2}(\frac{1}{1}+\frac{1}{27}) = 14/27 = 0.52$$

For Cut B

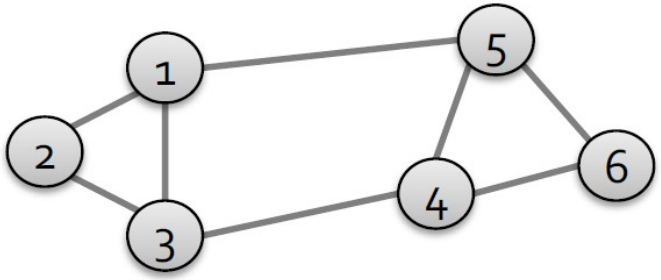
$$\text{Ratio Cut}(\{1,2,3,4\},\{5,6,7,8,9\}) = \frac{1}{2}(\frac{2}{4}+\frac{2}{5}) = 9/20 = 0.45 < 0.56$$

$$\text{Normalized Cut}(\{1,2,3,4\},\{5,6,7,8,9\}) = \frac{1}{2}(\frac{2}{12}+\frac{2}{16}) = 7/48 = 0.15 < 0.52$$

Both ratio cut and normalized cut prefer a balanced parti

More on Spectral Clustering

- **Laplacian matrix (L):**
  - $n \times n$ symmetric matrix



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

$$L = D - A$$

- **What is trivial eigenpair?**
  - $x = (1, \ldots, 1)$ then $L \cdot x = 0$ and so $\lambda = \lambda_1 = 0$
- **Important properties of L:**
  - **Eigenvalues** are non-negative real numbers
  - **Eigenvectors** are real (and always orthogonal)

Laplacian Matrix

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Proposition 1 (Properties of L) *The matrix L satisfies the following properties:*

1. *For every vector $f \in \mathbb{R}^n$ we have*

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

2. *L is symmetric and positive semi-definite.*

3. *The smallest eigenvalue of L is 0,*
   the corresponding eigenvector is the constant one vector $\mathbb{1}$.

4. *L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Laplacian Matrix

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Proof.
Part (1): By the definition of d_i,

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

Part (2): The symmetry of L follows directly from the symmetry of W and D. The positive semi-definiteness is a direct consequence of Part (1), which shows that $f'Lf \geq 0$ for all $f \in \mathbb{R}^n$.
Part (3): Obvious.
Part (4) is a direct consequence of Parts (1) - (3). □

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Let A be a linear transformation represented by a matrix $\mathbf{A}$. If there is a vector $\mathbf{X} \in \mathbb{R}^n \neq \mathbf{0}$ such

$$\mathbf{A}\mathbf{X} = \lambda \mathbf{X}$$

for some scalar λ, then λ is called the eigenvalue of $\mathbf{A}$ with corresponding (right) eigenvector $\mathbf{X}$.

Letting $\mathbf{A}$ be a $k \times k$ square matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}$$

with eigenvalue λ, then the corresponding eigenvectors satisfy

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix},$$

Positive Semidefinite Matrix

A positive semidefinite matrix is a Hermitian matrix all of whose eigenvalues are nonnegative.

Spectral Clustering

Reformulating ratio cut (or normalized cut) in matrix format

- Community membership matrix X; $X_{ij} = 1$, when node i is member of community j; 0, otherwise
- Let $D = diag(d_1, d_2, \cdots, d_n)$ be the diagonal degree matrix
- The ith entry on the diagonal of $X^T AX$ is the number of edges that are inside community i.
- The ith element on the diagonal of $X^T DX$ is the number of edges that are connected to members of community i.
- The ith element on the diagonal of $X^T(D - A)X$ is the number of edges in the cut that separates community i from other nodes.

The ith diagonal element of $X^T(D - A)X$ is equivalent to $\text{cut}(P_i, \overline{P_i})$

Spectral Clustering

So ratio cut is

$$\text{Ratio Cut}(P) = \frac{1}{k}\sum_{i=1}^{k}\frac{\text{cut}(P_i, \bar{P}_i)}{|P_i|}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\frac{X_i^T(D-A)X_i}{X_i^T X_i}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\hat{X}_i^T(D-A)\hat{X}_i$$

$$\hat{X}_i = X_i/(X_i^T X_i)^{1/2}$$

Spectral Clustering

Both ratio/normalized cut can be reformulated as

$$\min_{\hat{X}} \mathrm{Tr}(\hat{X}^T L \hat{X})$$

$$L = \begin{cases} D - A & \text{Ratio Cut Laplacian, i.e., Unnormalized Laplacian} \\ I - D^{-1/2} A D^{-1/2} & \text{Normalized Laplacian for Normalized Cut.} \end{cases}$$

$D = diag(d_1, d_2, \cdots, d_n)$ is a diagonal degree matrix

- It has been shown that both ratio cut and normalized cut minimization are NP-hard

NP-Hard

Approximation algorithms using relaxations are desired

Approximating RatioCut for k = 2

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

- The simplest case: partitioning the graph into k=2 subgraphs
- Our goal is to solve the optimization problem:

$$\min_{A \subset V} \mathrm{RatioCut}(A, \overline{A}).$$

Given a subset $A \subset V$ we define the vector $f = (f_1, \ldots, f_n)' \in \mathbb{R}^n$ with entries

$$f_i = \begin{cases} \sqrt{|\overline{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\overline{A}|} & \text{if } v_i \in \overline{A}. \end{cases}$$

Approximating RatioCut for k = 2

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

- Now the RatioCut objective function can be conveniently rewritten using the unnormalized graph Laplacian. This is due to the following calculation:

$$
\begin{aligned}
f'Lf &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{2}\sum_{i\in A, j\in \overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 + \frac{1}{2}\sum_{i\in \overline{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\
&= \mathrm{cut}(A,\overline{A})\left(\frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} + 2\right) \\
&= \mathrm{cut}(A,\overline{A})\left(\frac{|A|+|\overline{A}|}{|A|} + \frac{|A|+|\overline{A}|}{|\overline{A}|}\right) \\
&= |V| \cdot \mathrm{RatioCut}(A,\overline{A}).
\end{aligned}
$$

Approximating RatioCut for k = 2

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Additionally, we have

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i\in \overline{A}}\sqrt{\frac{|A|}{|\overline{A}|}} = |A|\sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}|\sqrt{\frac{|A|}{|\overline{A}|}} = 0.$$

In other words, the vector f as defined in Equation (2) is orthogonal to the constant one vector $\mathbb{1}$.

Finally, note that f satisfies

$$\|f\|^2 = \sum_{i=1}^{n} f_i^2 = |A|\frac{|\overline{A}|}{|A|} + |\overline{A}|\frac{|A|}{|\overline{A}|} = |\overline{A}| + |A| = n.$$

Approximating RatioCut for k = 2

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Altogether we can see that the problem of minimizing (1) can be equivalently rewritten as

$$\min_{A \subset V} f' L f \text{ subject to } f \perp \mathbb{1},\ f_i \text{ as defined in Eq. (2)},\ \|f\| = \sqrt{n}.$$

This is a discrete optimization problem as the entries of the solution vector *f* are only allowed to take two particular values, and of course it is still **NP hard**.

The most obvious relaxation in this setting is to discard the discreteness condition and instead allow that f_i takes arbitrary values in $\mathbb{R}$. This leads to the relaxed optimization problem

$$\min_{f \in \mathbb{R}^n} f' L f \text{ subject to } f \perp \mathbb{1},\ \|f\| = \sqrt{n}.$$

Generalize to k = N

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

The relaxation of the RatioCut minimization problem in the case of a general value *k* follows a similar principle as the one above.

Given a partition of V into k sets $A_1, \ldots, A_k$, we define k indicator vectors $h_j = (h_{1,j}, \ldots, h_{n,j})'$ by

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases} \qquad (i = 1, \ldots, n;\ j = 1, \ldots, k)$$

Then we set the matrix $H \in \mathbb{R}^{n \times k}$ as the matrix containing those k indicator vectors as columns. Observe that the columns in H are orthonormal to each other that is $H'H = I$.

Generalize to k = N

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Similar to the calculations when k=2, we can see that

$$h_i' L h_i = \frac{\mathrm{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

Moreover, one can check that

$$h_i' L h_i = (H' L H)_{ii}.$$

Combining those facts we get:

$$\mathrm{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} h_i' L h_i = \sum_{i=1}^{k} (H' L H)_{ii} = \mathrm{Tr}(H' L H),$$

where Tr denotes the trace of a matrix.

Generalize to k = N

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

So the problem of minimizing RatioCut($A_1, \ldots, A_k$) can be rewritten as

$$\min_{H \in \mathbb{R}^{n \times k}} \operatorname{Tr}(H'LH) \text{ subject to } H'H = I.$$

Because for any A which is a symmetric matrix:

$$\forall X \in \mathbb{R}^{n}, \ \lambda_{min} X^T X \leq X^T A X \leq \lambda_{max} X^T X$$

Then minimizing the trace is equivalent to find the K smallest eigenvalues and their corresponding eigenvectors.

That is where we connect the eigen-decomposition of Laplacian matrix and minimum cut.

Summary

Given L, the top k eigenvectors corresponding to the smallest eigen values are computed and denoted as $\hat{X}$.

Each node can be represented as a feature vector of dimension k.

Run k-means on these features to separate nodes into k classes/communities.

Note that the first eigenvector is meaningless (because it is zero and its corresponding eigenvector is a constant 1 vector); hence, the rest of the eigenvectors (k-1) are often used as k-means input.

Spectral Clustering Algorithms

From Miss Xin Xia' s tutorial
This won not be in your exam
If you feel difficult to understand, skip them.

Unnormalized spectral clustering

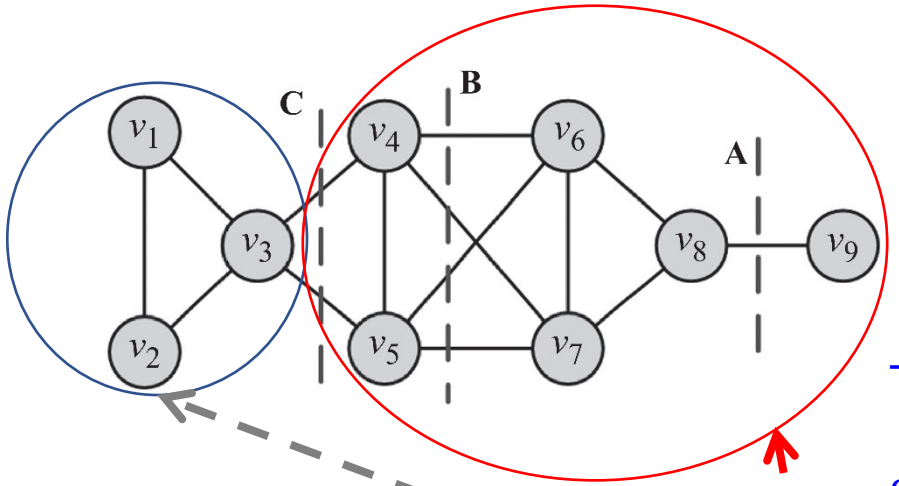
Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \, y_j \in C_i\}$.

Spectral Clustering: Example

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Two communities:
$\{v_1, v_2, v_3\}$
$\{v_4, v_5, v_6, v_7, v_8, v_9\}$

$$D = diag(2, 2, 4, 4, 4, 4, 4, 3, 1)$$

$$L = D - A = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 4 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

2 Eigenvectors
i.e., we want
2 communities

$$\begin{bmatrix} 0.33 & -0.46 \\ 0.33 & -0.46 \\ 0.33 & -0.26 \\ 0.33 & 1.16 \times 10^{-16} \\ 0.33 & 1.16 \times 10^{-16} \\ 0.33 & 0.13 \\ 0.33 & 0.13 \\ 0.33 & 0.33 \\ 0.33 & 0.59 \end{bmatrix}$$

k-means, $k = 2$

Summary of Spectral Clustering

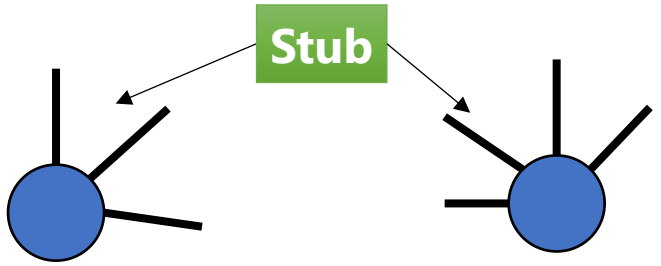
- Given L, the top k eigenvectors corresponding to the smallest eigenvalues are computed and denoted as $\hat{X}$.
  - Each node can be represented as a feature vector of dimension k.
- Run k-means on these features to separate nodes into k classes/communities.
- Note that the first eigenvector is meaningless (why?); hence, the rest of the eigenvectors (k-1) are often used as k-means input.

Community Detection

- Spectral Clustering
- Modularity Maximization

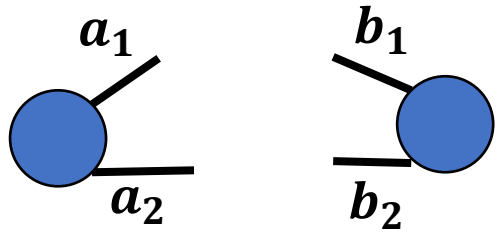
Modular Communities -Configuration Model

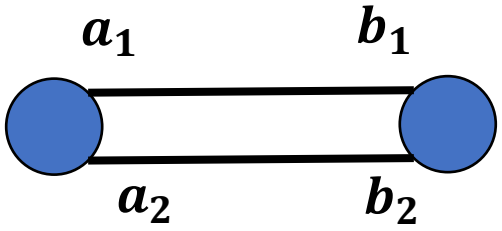
I. Given the degree sequence d_1, $d_2, \ldots, d_n$ generate n nodes, such that node i has d_i ***stubs*** (also known as ***half-edges***)

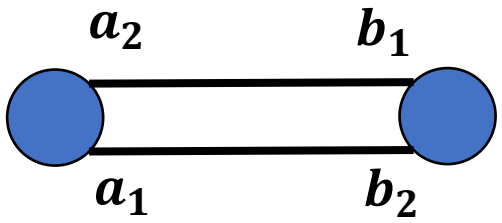


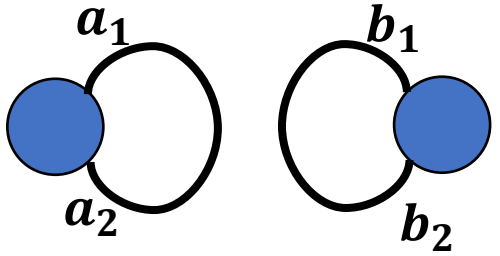
II. Randomly match stubs until there are no more stubs

- You can have loops and multiple edges
- Each **configuration** (**not each graph!**) appears with equal probability
- You can get the same graph









How to generate the Configuration model

1. Create a list where the **node id** for node v_i with degree d_i is repeated d_i times
2. Shuffle the list
3. Starting from the first index, join adjacent nodes

Example: Degree sequence (2,2,2)

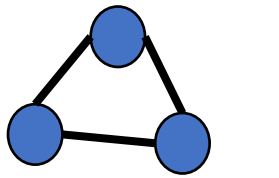
v_1	v_1	v_2	v_2	v_3	v_3

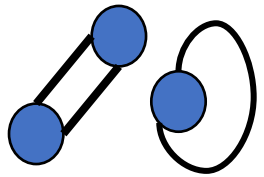
Random Shuffle 1:

v_1	v_2	v_2	v_3	v_3	v_1

Random Shuffle 2:

v_1	v_1	v_2	v_3	v_3	v_2





Modularity and Modularity Maximization

- Given a degree distribution, we know the expected number of edges between any pairs of vertices

- We assume that real-world networks should be far from random. Therefore, the more distant they are from this randomly generated network, the more structural they are.

- Modularity defines this distance and modularity maximization tries to maximize this distance

Normalized Modularity

Consider a partitioning of a graph $P = (P_1, P_2, P_3, \ldots, P_k)$

For partition P_x, this distance can be defined as

$$\sum_{i,j \in P_x} A_{ij} - \frac{d_i d_j}{2m}$$

This distance can be generalized for a partitioning P

$$\sum_{x=1}^{k} \sum_{i,j \in P_x} A_{ij} - \frac{d_i d_j}{2m}$$

The normalized version of this distance is defined as Modularity

$$Q = \frac{1}{2m} \sum_{x=1}^{k} \sum_{i,j \in P_x} A_{ij} - \frac{d_i d_j}{2m}$$

Modularity Maximization

Modularity matrix $B = A - dd^T/2m$

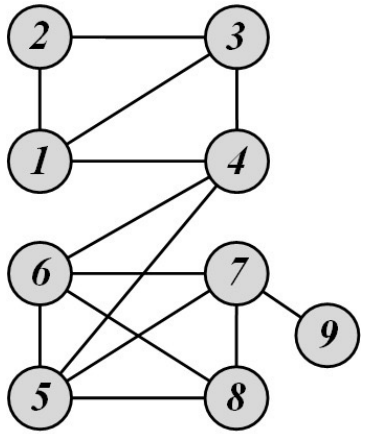
$d \in \mathbb{R}^{n\times 1}$ is the degree vector for all nodes

Reformulation of the modularity $Q = \frac{1}{2m}\mathrm{Tr}(X^T B X)$

- $X \in \mathbb{R}^{n\times k}$ is the indicator (partition membership) function:
  - $X_{ij} = 1$ iff. $v_i \in P_j$

- Similar to Spectral clustering,
  - We relax X to be real-valued matrix $\hat{X}$
  - The optimal solution for $\hat{X}$ is the top k eigenvectors of B.
  - To recover the original X, we can run k-means on $\hat{X}$

Modularity Maximization: Example



$$B = A - dd^T/2m$$
$$B_{ij} = A_{ij} - d_i d_j/2m$$

Two Communities:
{1, 2, 3, 4}
and
{5, 6, 7, 8, 9}

k-means

$$B = \begin{bmatrix} -0.32 & 0.79 & 0.68 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\ 0.79 & -0.14 & 0.79 & -0.29 & -0.29 & -0.29 & -0.29 & -0.21 & -0.07 \\ 0.68 & 0.79 & -0.32 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\ 0.57 & -0.29 & 0.57 & -0.57 & 0.43 & 0.43 & -0.57 & -0.43 & -0.14 \\ -0.43 & -0.29 & -0.43 & 0.43 & -0.57 & 0.43 & 0.43 & 0.57 & -0.14 \\ -0.43 & -0.29 & -0.43 & 0.43 & 0.43 & -0.57 & 0.43 & 0.57 & -0.14 \\ -0.43 & -0.29 & -0.43 & -0.57 & 0.43 & 0.43 & -0.57 & 0.57 & 0.86 \\ -0.32 & -0.21 & -0.32 & -0.43 & 0.57 & 0.57 & 0.57 & -0.32 & -0.11 \\ -0.11 & -0.07 & -0.11 & -0.14 & -0.14 & -0.14 & 0.86 & -0.11 & -0.04 \end{bmatrix}$$

Modularity Matrix

2 eigenvectors

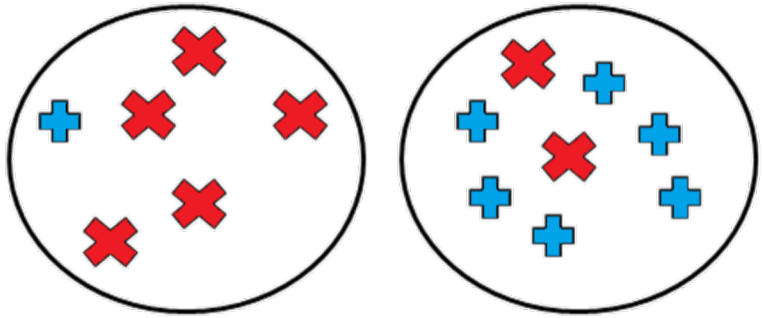
$$\begin{bmatrix} 0.44 & -0.00 \\ 0.38 & 0.23 \\ 0.44 & -0.00 \\ 0.17 & -0.48 \\ -0.29 & -0.32 \\ -0.29 & -0.32 \\ -0.38 & 0.34 \\ -0.34 & -0.08 \\ -0.14 & 0.63 \end{bmatrix}$$

Community Evaluation

Evaluating the Communities

We are given objects of two different kinds (+, ×)

- **The perfect community:** all objects inside the community are of the same type



- **Evaluation with ground truth**
- **Evaluation without ground truth**

[Xiangguo]:
This might be in your final exam

Precision and Recall

$$Precision = \frac{Relevant\ and\ retrieved}{Retrieved}$$

$$P = \frac{TP}{TP + FP}$$

$$Recall = \frac{Relevant\ and\ retrieved}{Relevant}$$

$$R = \frac{TP}{TP + FN}$$

True Positive (TP) :

- When similar member pairs are assigned to the same communities
- A **correct** decision.

False Negative (FN) :

- When similar member pairs are assigned to different communities
- An **incorrect** decision

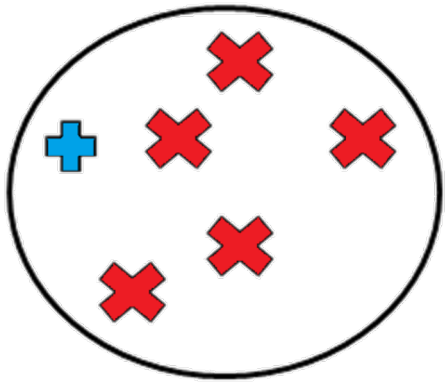
True Negative (TN) :

- When dissimilar member pairs are assigned to different communities
- A **correct** decision

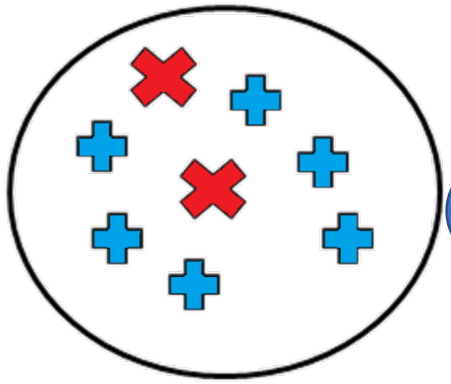
False Positive (FP) :

- When dissimilar member pairs are assigned to the same communities
- An **incorrect** decision

Precision and Recall: Example



Cluster 1



Cluster 2

[Xiangguo]:
This might be in your final exam

$$
\begin{aligned}
TP &= \binom{5}{2} + \binom{6}{2} + \binom{2}{2} = 26, \\
FP &= (5 \times 1) + (6 \times 2) = 17, \\
FN &= (5 \times 2) + (6 \times 1) = 16, \\
TN &= (6 \times 5) + (2 \times 1) = 32.
\end{aligned}
$$

$$P = \frac{26}{26+17} = 0.60$$

$$R = \frac{26}{26+16} = 0.61$$

F-Measure

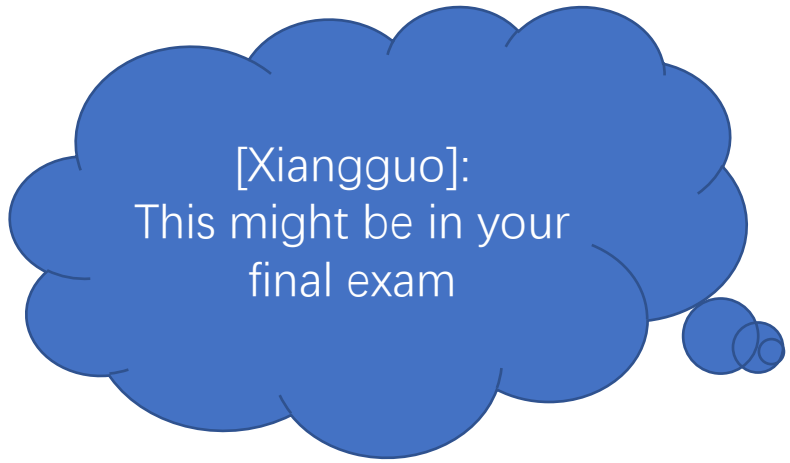
Either P or R measures one aspect of the performance,

- To integrate them into one measure, we can use the harmonic mean of precision of recall

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

For the example earlier,

$$F = 2 \times \frac{0.6 \times 0.61}{0.6 + 0.61} = 0.60$$



Purity

[Xiangguo]:
This might be in your final exam

We can assume the majority of a community represents the community

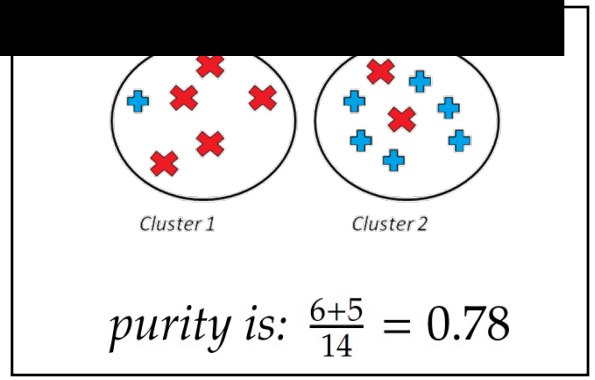
- We use the label of the majority against the label of each member to evaluate the communities

Purity. The fraction of instances that have labels equal to t

Purity can be easily **tampered** by

- Points being singleton communities (of size 1); or by
- Very large communities

$$Purity = \frac{1}{N}\sum_{i=1}^{k}\max_{j}|C_i \cap L_j|$$



- k: the number of communities
- N: total number of nodes,
- L_j: the set of instances with label j in all communities
- C_i: the set of members in community i

Mutual Information

- **Mutual information (MI).** The amount of information that two random variables share.
  - By knowing one of the variables, it measures the amount of uncertainty reduced regarding the others

$$MI = I(H, L) = \sum_{h \in H} \sum_{l \in L} \frac{n_{h,l}}{n} \log \frac{n \cdot n_{h,l}}{n_h n_l}$$

- L and H are labels and found communities;
- n_h and n_l are the number of data points in community h and with label l, respectively;
- $n_{h,l}$ is the number of nodes in community h and with label l; and n is the number of nodes

Normalizing Mutual Information (NMI)

- Mutual information (MI) is unbounded
- To address this issue, we can normalize MI

- How? We know that

$$MI \leq min(H(L), H(H)),$$
$$(MI)^2 \leq H(H)H(L).$$
$$MI \leq \sqrt{H(H)}\sqrt{H(L)}.$$

- $H(.)$ is the entropy function

$$H(L) = -\sum_{l \in L} \frac{n_l}{n} \log \frac{n_l}{n}$$
$$H(H) = -\sum_{h \in H} \frac{n_h}{n} \log \frac{n_h}{n}.$$

Normalized Mutual Information

$$NMI = \frac{MI}{\sqrt{H(L)}\sqrt{H(H)}}.$$

$$NMI = \frac{\sum_{h \in H} \sum_{l \in L} n_{h,l} \log \frac{n \cdot n_{h,l}}{n_h n_l}}{\sqrt{(\sum_{h \in H} n_h \log \frac{n_h}{n})(\sum_{l \in L} n_l \log \frac{n_l}{n})}}.$$

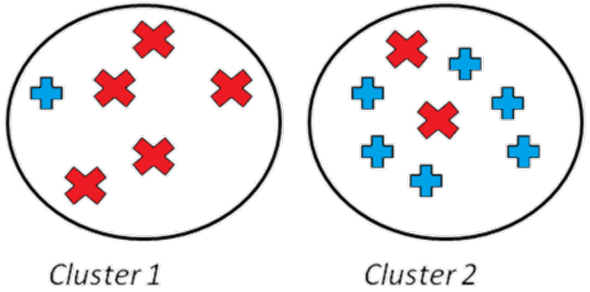
Normalized Mutual Information

$$NMI = \frac{\sum_{h,l} n_{h,l} \log \frac{n \cdot n_{h,l}}{n_h n_l}}{\sqrt{(\sum_h n_h \log \frac{n_h}{n})(\sum_l n_l \log \frac{n_l}{n})}}$$

- where l and h are known (with labels) and found communities, respectively
- n_h and n_l are the number of members in the community h and l, respectively,
- $n_{h,l}$ is the number of members in community h and labeled l,
- n is the size of the dataset

- **NMI** values close to one indicate high similarity between communities found and labels
- Values close to zero indicate high dissimilarity between them

Normalized Mutual Information: Example



[Xiangguo]:
This might be in your final exam

Found communities (H)

- [1,1,1,1,1,1, 2,2,2,2,2,2,2,2]

Actual Labels (L)

- [2,1,1,1,1,1, 2,2,2,2,2,2,1,1]

$n = 14$

	$\mathbf{n}_h$
h=1	6
h=2	8

	$\mathbf{n}_l$
$l = 1$	7
$l = 2$	7

$\mathbf{n}_{h,l}$	$l = 1$	$l = 2$
h=1	5	1
h=2	2	6