باسمه تعالی

دانشگاه شریف

دانشکده برق

فاز دوم پروژه درس

آمار واحتمال مهندسی

دانشجویان:

محمدپارسا دینی

محمدمتین میرزابابایی

مصطفی آذری

دکتر کرباسی

بهمن 1401

بخش۱ : من از دیار حبیبم نه از بلاد غریب!

پرسش تئوری ۱

از آنجایی که ماتریس کاملا متقارن است فاصله میانگین هر راس برابر با فاصله میانگین گراف است. برای یک راس فاصله میانگین را محاسبه میکنیم به این صورت که از سمت چپ و راست mتا mتا از راس ها جدا میکنیم تا جایی که r راس باقی بماند که r باقی مانده $n-1$ به $2m$ است. (به این خاطر که از دو طرف mتا mتا برمیداریم.) $2k$ دسته داریم. برابر است با:

$$k = \left\lfloor \frac{n-1}{2m} \right\rfloor$$

این $2k$ راس را به این صورت نامگذاری میکنیم که راس های سمت راست آن را از ۱ تا m و راسهای سمت چپ آن را از -۱ تا $-m$ در نظر میگیریم. برای راسهای درون دسته iام کمینه فاصله برابر با اندازه i است. کمینه فاصله برای نقاط درون دسته r برابر با $k+1$ است.

بنابراین فاصله میانگین برای یک راس برابر میشود با:

$$d_{mean} = 2m \sum_{i=-k}^{+k} |i| + r(k+1) = 2mk(k+1) + r(k+1) = (2mk+r)(k+1)$$

بنابراین فاصله میانگین برابر است با:

$$d_{mean} = (2mk+r)(k+1)$$

که در آن :

$$k = \left\lfloor \frac{n-1}{2m} \right\rfloor$$

$$r = n - 1 - 2m \left\lfloor \frac{n-1}{2m} \right\rfloor$$

پرسش تئوری ۲

ضریب خوشه بندی یک رأس، به صورت نسبت تعداد یال های بین رأس های همسایه ی آن رأس به تعداد تمام یال هایی که در گراف کامل (یعنی گرافی که همه ی یال ها در آن وجود دارند) بین همین همسایه ها وجود دارد، تعریف می شود. یال های بین رأس های همسایه ی آن رأس به صورت ریاضی به شکل زیر بیان میشود:

$$|\{e_{i,j}:\ v_k, v_j \in N_i, e_{k,j} \in E\}|$$

و همچنین تعداد تمام یال هایی که در گراف کامل برابر است با

$$\frac{1}{\frac{k_i(k_i-1)}{2}}$$

بنابراین ضریب خوشه بندی یک رأس برابر میشود با :

$$C_i = \frac{2|\{e_{i,j}:\ v_k, v_j \in N_i, e_{k,j} \in E\}|}{k_i(k_i-1)}$$

پرسش تئوری ۳

مجموعه راس های همسایه یک راس در گراف $g(n,m)$ برابر m راس سمت چپ و m راس سمت راست آن است. این $2m$ راس را به این صورت نامگذاری میکنیم که راس های سمت راست آن را از ۱ تا m و راسهای سمت چپ آن را از -۱ تا $-m$ در نظر میگیریم. برای راس iام تعداد یال های بین آن و دیگر همسایه ها برابر است با:

$$n_i = 2m - |i|$$

تعداد یال های بین راس های همسایه ی آن راس برابر میشود با:

$$N = \frac{1}{2}\sum_{i=-m}^{+m} n_i - 2m = \frac{2m(2m+1)}{2} - \frac{m(m+1)}{2} - 2m = \frac{3}{2}m(m-1)$$

تعداد تمام یال هایی که در گراف کامل (یعنی گرافی که همه ی یال ها در آن وجود دارند) بین همین همسایه ها وجود دارد برابر است با:

$$N_T = \frac{2m(2m-1)}{2} = m(2m-1)$$

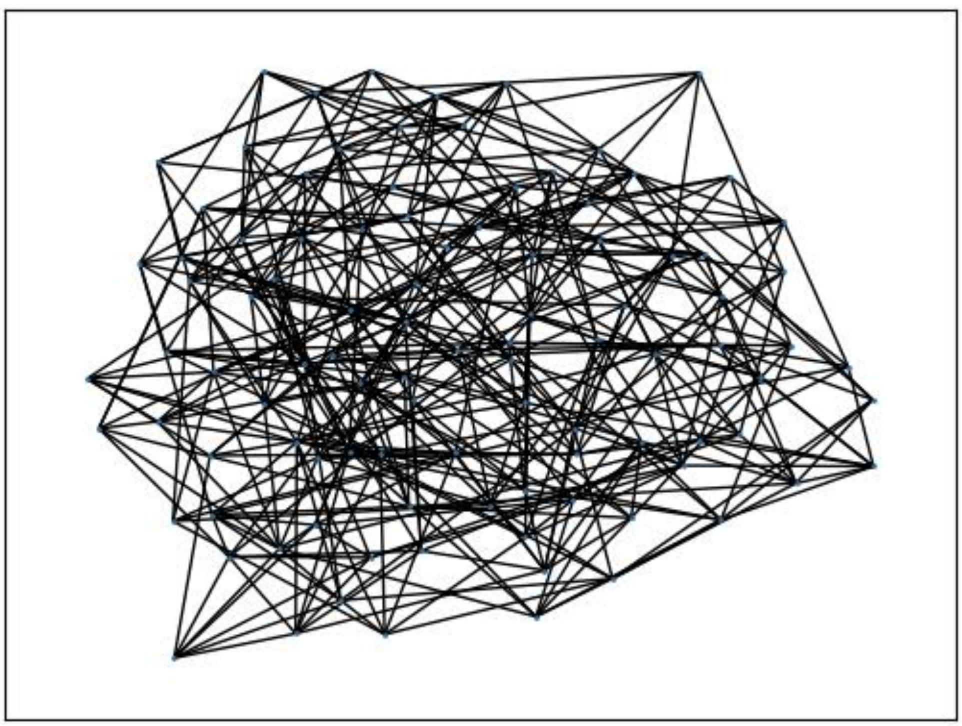
از آنجایی که ماتریس کاملا متقارن است ضریب خوشه بندی هر راس برابر با ضریب خوشه بندی میانگین است. بنابراین ضریب خوشه بندی میانگین برابر میشود با:

$$r = \frac{3(m-1)}{2(2m-1)}$$

که تنها در حالت برابر با $\frac{1}{2}$ میشود که $m = 2$ باشد.

پرسش شبیه سازی ۱

این گراف به صورت آماده در لایبرری $networkx$ وجود دارد. برای تولید این گراف از این دستور استفاده میکنیم. به ازای $n = 10000$ و $m = 400$ گراف را نمیتوان رسم کرد. اما به ازای $n = 100$ و $m = 4$ گراف به صورت زیر است:



پرسش تئوری ۴

بردار d به اندازه d_i تا درایه i دارد که این تعداد درون تعدادی دوتایی (i, j) قرار دارند. بنابراین برای به دست آوردن d_i میتوانیم تعداد دوتایی هایی که یکی از دو مقدار آنها i است را محاسبه کنیم. داریم:

$$d_i = |\{(\mathrm{j}, \mathrm{k}) : \mathrm{j} == \mathrm{i} \text{ or } \mathrm{k} == \mathrm{i}\}|$$

دو دسته دوتایی در این مجموعه قرار میگیرند. یکی آنهایی که به شکل (i,j) هستند و تعداد آنها $A_{i,j}$ است و یک نوع که تعداد (i,i) آن $A_{i,i}$ است و باید دوباره حساب شود چون دوتا i دارد. بنابراین d_i برابر با مقدار زیر میشود:

$$d_i = A_{i,i} + \sum_{i=1}^{n} A_{i,j}$$

پرسش تئوری ۵

مقدار d دو برابر m است چون mتا دوتایی داریم که در مجموع تعداد اعضا این دوتایی ها برابر d است. بنابراین داریم:

$$m = \frac{1}{2}d = \frac{1}{2}\sum_{i=1}^{n} d_i$$

پرسش تئوری ۶

اعضا مجموعه d را پشت سر هم به صورت دنباله می چینیم و سپس به ترتیب از سمت ارست دوتا دوتا جدا میکنیم. به این صورت ماتریس $\hat{A}$ را میتوان تشکیل داد. برای به دست آوردن احتمال مشاهده ماتریس $\hat{A}$ درصورتی که d را داشته باشیم، باید تعداد دنباله هایی که به ازای آنها ماتریس حاصل $\hat{A}$ خواهد بود (N_f) را تقسیم بر کل دنباله های ممکن (N_A) کنیم.

$$P\left[A = \hat{A}\middle|d\right] = \frac{N_f}{N_A}$$

تعداد کل دنباله های ممکن برابر است با:

$$N_A = 2m!$$

تعداد حالات مطلوب برابر با تعداد جایگشت های دنباله d است به صورتی که ماتریس حاصل $\hat{A}$ بماند. تعداد کل جایگشت ها ناشی از سه نوع جایگشت است:

۱_ از هر رقم i به تعداد d_i داریم که $d_i!$ جایگشت دارد. بنابراین تعداد این نوع جایگشت برابر میشود با:

$$N_1 = \prod_{i=1}^{n} d_i!$$

۲_ دوتایی ها را میتوان جابجا کرد بدون اینکه ماتریس حاصل تغییر کند. اما در این جایگشت ها زمانی که دو دوتایی یکسان را جابجا کنیم، عملا مانند این است که مکان دو دوتایی را ثابت نگه داشته و جایگشت اعضا آن دو را در نظر بگیریم که همه این حالات در جایگشت نوع یک لحاظ شده است. بنابراین باید آنها در این جایگشت در نظر نگیریم.

با این توصیفات تعداد این نوع جایگشت برابر است با:

$$N_2 = \frac{m!}{\prod_{0<i\le j\le n}^{n} A_{i,j}!}$$

۳_ در هر دوتایی (i,j) جایگشتی به صورت (j,i) دارد در صورتی که $i = j$ نباشد چون در صورت برابر بودن این دو، نوع جایگشت از نوع یک لحاظ شده است. بنابراین تعداد جایگشت ها برابر میشود با:

$$N_3 = 2^{m-\sum_{i=1}^{n}\hat{A}_{i,i}}$$

که در آن $m - \sum_{i=1}^{n} A_{i,i}$ تعداد دوتایی هایی است که در آن دو عضو یکسان نیستند. میتوان به صورت زیر نیز نوشت:

$$N_3 = \frac{2^m}{2^{\sum_{i=1}^{n}\hat{A}_{i,i}}} = \frac{2^m}{\prod_{i=1}^{n} 2^{\hat{A}_{i,i}}}$$

سه نوع جایگشت به این صورت هستند. بنابراین تعداد دنباله هایی که به ازای آنها ماتریس حاصل $\hat{A}$ خواهد بود (N_f) برابر است با:

$$N_f = N_1 N_2 N_3$$

حال میتوانیم احتمال مشاهده ماتریس $\hat{A}$ درصورتی که d را داشته باشیم، برابر میشود با:

$$P\left[A=\hat{A}\middle|d\right]=\frac{N_f}{N_A}=\frac{1}{2m!}\frac{m!}{\prod_{0<i\le j\le n}A_{i,j}!}\frac{2^m}{\prod_{i=1}^{n}2^{\hat{A}_{i,i}}}\prod_{i=1}^{n}d_i!=\frac{2^m m!}{2m!}\frac{\prod_{i=1}^{n}d_i!}{\prod_{0<i\le j\le n}A_{i,j}!\ \prod_{i=1}^{n}2^{\hat{A}_{i,i}}}$$

کسر اول را میتوان به صورت زیر ساده کرد:

$$\frac{2^m m!}{2m!}=\frac{2^m\prod_{i=1}^{m}i}{\prod_{i=1}^{2m}i}=\frac{\prod_{i=1}^{m}2i}{\prod_{i=1}^{2m}i}=\frac{\prod_{i=1}^{m}2i}{\prod_{i=1}^{m}2i\prod_{i=1}^{m}(2i-1)}=\frac{1}{\prod_{i=1}^{m}(2i-1)}$$

بنابراین احتمال به صورت زیر ساده میشود:

$$P\left[A=\hat{A}\middle|d\right]=\frac{1}{\prod_{i=1}^{m}(2i-1)}\frac{\prod_{i=1}^{n}d_i!}{\prod_{0<i\le j\le n}A_{i,j}!\ \prod_{i=1}^{n}2^{\hat{A}_{i,i}}}$$

پرسش شبیه سازی ۴

اگر یک یال را به تصادف انتخاب کنیم احتمال اینکه طوقه یا ابریال نباشد برابر است با:

$$p=0.997$$

بنابراین وجود طوقه و ابر یال مشکلی به وجود نمی آورد چون احتمال اتفاق افتادن آن بسیار کم و تقریبا برابر با صفر است.

پرسش تئوری ۷

طبق توضیحات سوال، درایه $A_{i,j}$ یک متغییر تصادفی که به صورت برنولی با احتمال $p_{i,j}$ بیان میشود.

$$A_{i,j}\sim Bernoulli(p_{i,j})$$

از طرفی تعداد درجه راس v_i برابر با مجموع تعداد یال هایی است که یک سر آنها راس v_i باشد. بنابراین:

$$D_i=\sum_{j=1}^{n}A_{i,j}$$

در نتیجه

$$E[D_i]=\sum_{j=1}^{n}E[A_{i,j}]=\sum_{j=1}^{n}p_{i,j}$$

طبق توضیحات، $p_{i,j}$ به صورت زیر است:

$$p_{i,j}=\frac{d_i d_j}{\sum_{k=1}^{n}d_k}$$

با جایگذاری این مقدار خواهیم داشت:

$$E[D_i]=\sum_{j=1}^{n}p_{i,j}=\sum_{j=1}^{n}\frac{d_i d_j}{\sum_{k=1}^{n}d_k}=\frac{d_i}{\sum_{k=1}^{n}d_k}\sum_{j=1}^{n}d_j$$

بنابراین داریم:

$$E[D_i]=d_i$$

پرسش شبیه سازی ۵

مقادیر پیشنهادی را $n = 20000$ و $p = 400$ در نظر میگیریم که هم به مقادیر واقعی نزدیک و هم محاسبات آن بیش از حد زمان بر نباشد. تعداد یال و طول d مورد انتظار برابر است با:

$$m = \frac{nd}{2} = 4000000, \qquad |d| = nd = 8000000$$

مقادیر به دست آمده از طریق شبیه سازی هستند با:

$$m = 4000024.5, \qquad |d| = 4000099$$

پرسش تئوری ۸

احتمال همسلیقگی بین هر دو نفر یک متغییر برنولی است و از دیگر روابط همسلیقگی مستقل است. بنابراین اگر تعداد داده های خیلی زیاد باشد مجموع این چند متغییر برنولی به پوآسون میل میکند. امید ریاضی تعداد همسلیقگی برای هر فرد k است. درنتیجه از آنجایی که در توزیع پوآسون، پارمتر توزیع برابر امید ریاضی متغییر است، میتوان امید ریاضی تعداد همسلیقگی برای هر فرد را میتوان به صورت زیر در نظر گرفت:

$$D \sim poisson(k)$$

بنابراین احتمال آن که تعداد روابط هم سلیقگی هر کاربر در گراف تصادفی برابر k باشد برابر است با:

$$P[D = k] = \frac{k^k e^{-k}}{k!}$$

به ازای $k = 1$ داریم:

$$P[D = 1] = e^{-1} \leq \frac{1}{2}$$

نسبت $P[D = k]$ به ازای k و $k + 1$ برابر است با:

$$\frac{P[D = k]}{P[D = k + 1]} = \frac{\frac{(k + 1)^{k+1} e^{-k-1}}{(k + 1)!}}{\frac{k^k e^{-k}}{k!}} = \frac{e^{-k-1}}{e^{-k}} \frac{k!}{(k + 1)!} \frac{(k + 1)^{k+1}}{k^k} = e^{-1} \frac{1}{k + 1} \frac{(k + 1)^{k+1}}{k^k}$$
$$= e^{-1} \frac{(k + 1)^k}{k^k} = e^{-1} \left(1 + \frac{1}{k}\right)^k$$

با استفاده از نامساوی $e^{+k} \geq 1 + k$ داریم:

$$\frac{P[D = k]}{P[D = k + 1]} = e^{-1} \left(1 + \frac{1}{k}\right)^k \leq e^{-1} \left(e^{\frac{1}{k}}\right)^k = e^{-1} e^1 = 1$$

در نتیجه

$$\frac{P[D = k]}{P[D = k + 1]} \leq 1$$

بنابراین $P[D = k]$ نزولی خواهد بود.

از آنجایی که $P[D = k]$ نزولی است بیشیه مقدار خود را در کمترین مقدار k دارد که برابر یک است و به ازای این مقدار از ۰.۵ کمتر بود. بنابراین $P[D = k]$ همواره کمتر از $\frac{1}{2}$ خواهد بود.

پرسش تئوری ۹

در صورتی که $D_i = 0$ باشد، رابطه زیر باید برقرار باشد:

$$\forall\, j \in n: \qquad A_{i,j} = 0$$

از آنجایی که وجود و عدم وجود یال های مختلف از هم مستقل است میتوان به صورت زیر نوشت:

$$P[D_i = 0] = P\left[\forall\, j \in n:\ A_{i,j} = 0\right] = P\left[A_{i,1} = 0\right]P\left[A_{i,2} = 0\right] \dots P\left[A_{i,n} = 0\right]$$
$$= \left(1 - p_{i,1}\right)\left(1 - p_{i,2}\right) \dots \left(1 - p_{i,n}\right)$$

از نامساوی $e^{-k} \geq 1 - k$ استفاده میکنیم. داریم:

$$P[D_i = 0] = \left(1 - p_{i,1}\right)\left(1 - p_{i,2}\right) \dots \left(1 - p_{i,n}\right) \leq e^{-p_{i,1}} e^{-p_{i,2}} \dots e^{-p_{i,2}} = e^{-\sum_{j=1}^{n} p_{i,j}}$$

از پرسش تئوری ۷ داریم:

$$\sum_{j=1}^{n} p_{i,j} = d_i$$

در نتیجه داریم:

$$P[D_i = 0] \leq e^{-d_i}$$

پرسش تئوری ۱۰

احتمال اینکه درجه ی تمام رئوس بزرگ تر یا مساوی با ۱ باشد، متمم این است که حداقل درجه یک راس درجه صفر باشد. بنابراین میتوان به صورت زیر محاسبه کرد:

$$P = 1 - P[\exists i \in n:\ D_i = 0] \geq 1 - P[D_1 = 0] - P[D_2 = 0] \dots P[D_n = 0] = 1 - \sum_{i=1}^{n} P[D_i = 0]$$

از نتیجه پرسش قبل داریم:

$$P \leq 1 - \sum_{i=1}^{n} e^{-d_i}$$

بنابراین اگر $\sum_{i=1}^{n} e^{-d_i} \leq \varepsilon$ باشد به احتمال $1 - \varepsilon$ درجه تمامی راس ها بزرگتر یا مساوی یک است.

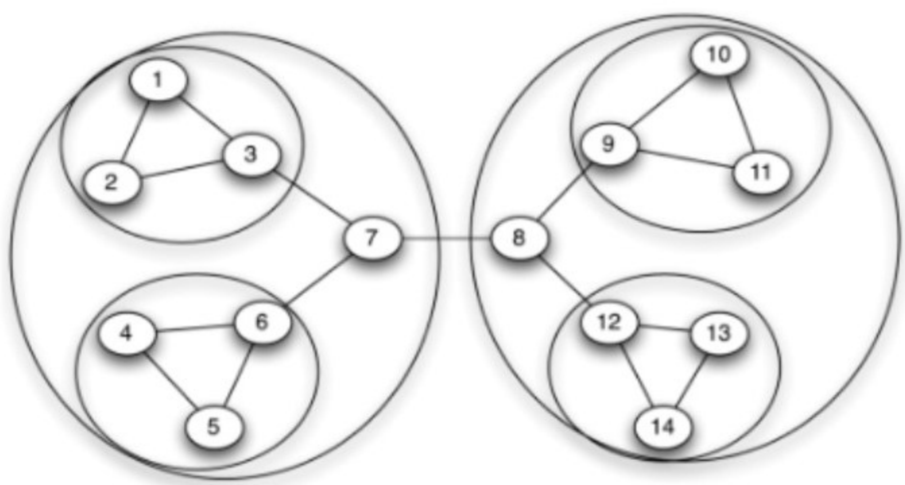
In the Name of God

Probability and Statistics Project

Phase2 Section2

$Theory\ Section\ 11:$

Distant relatives for sure. Consider this graph as a model describing the relationship between some people:



Assume that:

1: the person,

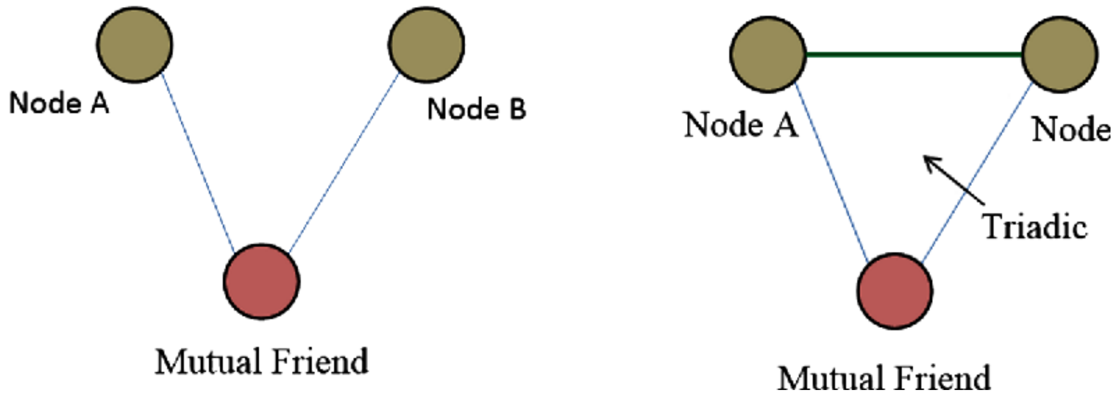
3: his father, 7: his uncle, 8: his uncle's friend, 9-14 : family of his uncle's friend,

Since the person visits his father and his uncle frequently, they are considered as his close family but his uncle's friend and his family are considered as distant relatives.For example, If we assume that the person is at the center of a circle, his connection would be proportional to R^2 and will increase as his connection radix R increases. So the news,job offer or ... are more likley to appear at the edge of circle due to its probability.

$Theory\ Section\ 12:$

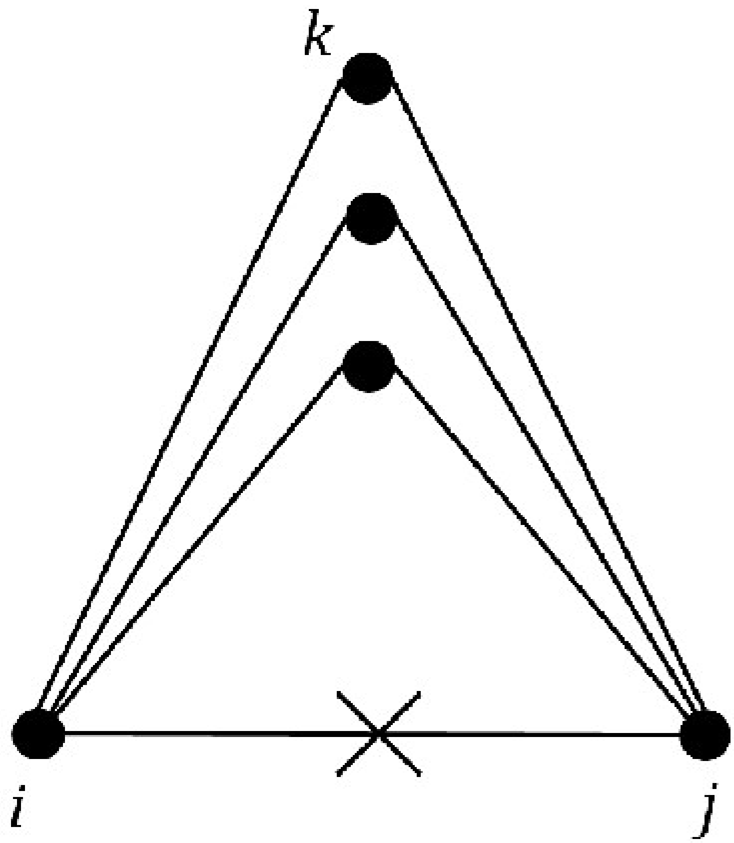
If the connections A-B and A-C exist, there is a tendency for the new connection B-C to be formed. Triadic closure can be used to understand and predict the growth of networks. In social networks, triadic closure facilitates cooperative behavior, but when new connections are made via referrals from existing connections the average global fraction of cooperators is less than when individuals choose new connections randomly from the population at large. Two possible effects for this are by the structural and informational construction. The structural construction arises from the propensity toward high clusterability. The informational construction comes from the assumption that an individual knows something about a friend's friend, as opposed to a random stranger.

. *Triadic closure property*



$Theory\ Section\ 13:$

Suppose that B and C are two people having common interests with people $A_1, A_2, \ldots, A_k$



$$P(A_1 \cup A_2 \cup A_3 \cup \ldots A_k) = (By\ inclusion)$$
$$P(A_1) + P(A_2) + P(A_3) + \ldots + P(A_k) - [P(A_1 \cap A_2) + P(A_1 \cap A_3) + \ldots] + [P(A_1 \cap A_2$$

Assuming Pbc = P : $\binom{k}{1}p - \binom{k}{2}p^2 + \binom{k}{3}p^3 - \ldots + (-1)^{k+1}p^k = \sum_{i=1}^{k} \binom{k}{i} p^i (-1)^{i+1}$

$$ifK = 1:$$

$$P(A_1 \cup A_2 \cup A_3 \cup \ldots A_k) = p$$

$$ifK = 2:$$

$$P(A_1 \cup A_2 \cup A_3 \cup \ldots A_k) = 2p - p^2$$

$$ifK = 3:$$

$$P(A_1 \cup A_2 \cup A_3 \cup \ldots A_k) = 3p - 3p^2 + p^3$$

$$ifK = 4:$$

$$P(A_1 \cup A_2 \cup A_3 \cup \ldots A_k) = 4p - 6p^2 + 4p^3 - p^4$$

For example the sequence for p = 0.5 probability would be: 0.5 , 0.75 , 0.875 , 0.9375 , ...

As we can see, the probability increases as k increases.

بخش۳: فرض ایزد بگزاریم و به کس بد نکنیم!

پرسش تئوری ۱۴

فرض مقابل را اینکه سلیقه یک خانم به یک خانم شبیه تر است تا یک آقا در نظر میگیریم. به صورت ریاضی فرض صفر و مقابل را می توان به صورت زیر نوشت:

$$H_0: \quad P_1 \le P_{12}$$

$$H_1: \quad P_1 > P_{12}$$

پرسش تئوری ۱۵

X_1 را متغییر تصادفی نشان دهنده ارتباط دو خانم و X_2 را متغییر تصادفی نشان دهنده ارتباط یک خانم و آقا در نظر میگیریم. متغییر تصادفی X و W را به صورت زیر تعریف میکینم:

$$X = X_1 - X_2$$
$$W = \frac{X}{\frac{S}{\sqrt{n}}}$$

برای خطای نوع اول داریم:

$$P[X_1 - X_2 > t | H_0] = 1 - \alpha$$

بر حسب W داریم:

$$P\left[\frac{X_1 - X_2}{\frac{S}{\sqrt{n}}} > \frac{t}{\frac{S}{\sqrt{n}}} | H_0\right] = 1 - \alpha$$

$\frac{t}{\frac{S}{\sqrt{n}}}$ را متغییر نرمال در نظر میگیریم.

$In\ the\ Name\ of\ God$

$Probability\ and\ Statistics\ Project$

$Phase2\ Section4$

$Theory\ Section\ 16:$

No, in the first iteration, at the first step, vertices 0 and 4 are not connected to each other(in the neighbourhood) and we are not allowed to connect them in any way.Except that, other steps are done according to Louvain algorithm. Note that when convergence is completed, the final state of the graph will be the same but at least in the first iterations it is not like that.

$Theory\ Section\ 17:$

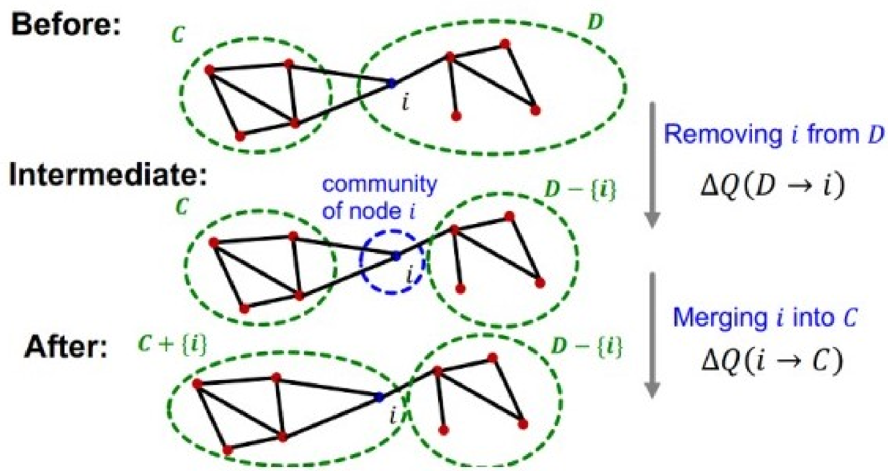
Firstly, we need to rememmber what the main aim of Louvain algorithm was. To summarize, In the Louvain Method of community detection, first small communities are found by optimizing modularity locally on all nodes, then each small community is grouped into one node and the first step is repeated. Since the overall weight of system is conservative and we have less edges, loop edges's weight increases(k_i, k_j decreases). On the other hand A_{ij} doesn't change or increases.Thus, the inside term of the below sum increases:
$Q = \frac{1}{2m}\sum_{i,j\in C}[A_{ij} - \frac{k_i k_j}{2m}]$

Accordingly, Q increases and it's pleasant!

$Theory\ Section\ 18:$

To move v_i to the cluster containing v_j first, we have to bring i out of D and then merg it into C.

In this section, we are asked to calculate just on of the transitions. ($\Delta Q(i->C)$)



شکل ۸: مثالی از جابجایی رئوس بین خوشه‌ها

$$Q(c) = \frac{1}{2m} \sum_{i,j \in C} [A_{ij} - \frac{k_i k_j}{2m}] = \frac{\sum_{i,j \in C} A_{ij}}{2m} - \frac{(\sum_{i \in C} k_i)(\sum_{i \in C} k_i)}{(2m)^2}$$

$$Q(c) = \frac{\sum in}{2m} - (\frac{\sum tot}{2m})^2 \quad where : \sum_{in} = \sum_{i,j \in c} A_{ij}, \sum_{tot} = \sum_{i \in c} k_i$$

$$Q_{before} = Q(c) + Q(i) = [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2] + [0 - (\frac{k_i}{2m}^2)]$$

$$Q_{after} = Q(c+i) = \frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2$$

$$\Delta Q(i- > c) = Q_{after} - Q_{before} = \frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2 - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 + [0 - (\frac{k_i}{2m}^2)]]$$

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

- A_{ij} represents the edge weight between nodes i and j;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively;
- $2m$ is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is an indicator function $\delta(c_i, c_j) = 1 \; if \; c_i = c_j \; else \; 0$

$Simulating\ Section\ 9:$

In [ ]:

```
import networkx as nx
import community
#from communities.algorithms import louvain_method
```

```
import matplotlib.cm as cm
import matplotlib.pyplot as plt

Graph = nx.karate_club_graph() #  karate club graph
partition = community.best_partition(Graph)

pos = nx.spring_layout(Graph)
cmap = cm.get_cmap('viridis', max(partition.values()) + 1)
nx.draw_networkx_nodes(Graph, pos, partition.keys(), node_size=60, cmap=cmap, node_
nx.draw_networkx_edges(Graph, pos, alpha=0.6)
plt.show()
```

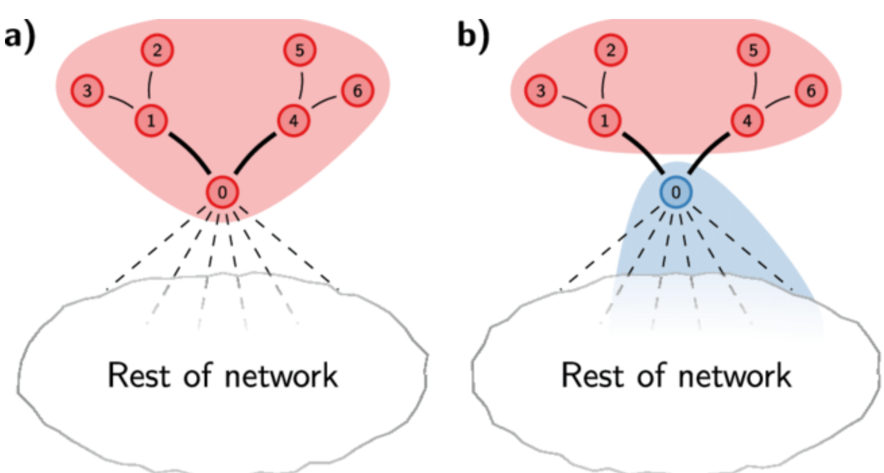
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[11], line 8
      5 import matplotlib.pyplot as plt
      7 Graph = nx.karate_club_graph() #  karate club graph
----> 8 partition = community.best_partition(Graph)
     10 pos = nx.spring_layout(Graph)
     11 cmap = cm.get_cmap('viridis', max(partition.values()) + 1)

AttributeError: module 'community' has no attribute 'best_partition'
```

Unfortunately, The installed library doesn't have this attribute.

For modularity: $$ modularity_matrix(adj_matrix : numpy.ndarray) -> numpy.ndarray Computes the modularity matrix for a graph.

$Simulating\ Section\ 10:$

Consider the partition shown in (a). When node 0 is moved to a different community, the red community becomes internally disconnected, as shown in (b). However, nodes 1-6 are still locally optimally assigned, and therefore these nodes will stay in the red community.



Importantly, the problem of disconnected communities is not just a theoretical curiosity.The problem occurs frequently in practice when using the Louvain algorithm. Perhaps surprisingly, iterating the algorithm aggravates the problem, even though it does increase the quality function. In the Louvain algorithm, a node may be moved to a different community while it may have acted as a bridge between different components of its old community. Removing such a node from its old community disconnects the old community. One may expect that other nodes in the old community will then also be moved to other communities. However, this is not necessarily the case, as the other nodes may still be sufficiently strongly connected to their community, despite the fact that the community has become disconnected.

This problem is different from the well-known issue of the resolution limit of modularity. Due to the resolution limit, modularity may cause smaller communities to be clustered into larger communities. In other words, modularity may “hide” smaller communities and may yield communities containing significant substructure.(Helped by article "From Louvain to Leiden: guaranteeing well-connected communities.")