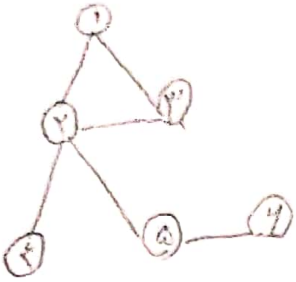


الف)

degree

$k_{۱} = ۲$ $k_{۲} = ۴$ $k_{۳} = ۲$ $k_{۴} = ۱$ $k_{۵} = ۲$ $k_{۶} = ۱$

closeness centrality

$$C_v = \frac{N-1}{\sum\limits_{u \neq v} \text{shortest path length between } u \text{ and } v}$$

	۱	۲	۳	۴	۵	۶
۱	۰	۱	۱	۲	۲	۳
۲	۱	۰	۱	۱	۱	۲
۳	۱	۱	۰	۲	۲	۳
۴	۲	۱	۲	۰	۲	۳
۵	۲	۱	۲	۲	۰	۱
۶	۳	۲	۳	۳	۱	۰

۱ + ۱ + ۲ + ۲ + ۳ = ۹

$C_{۱} = \frac{۶-۱}{۹} = ۰/۵۵۵۵$ (۱-۲, ۱-۳, ۱-۲-۴, ۱-۲-۵, ۱-۲-۵-۶)

$C_{۲} = \frac{۵}{۶} = ۰/۸۳۳۳$ (۲-۱, ۲-۳, ۲-۴, ۲-۵, ۲-۵-۶)

۱ + ۱ + ۱ + ۱ + ۲ = ۶

$C_{۳} = \frac{۵}{۹} = ۰/۵۵۵۵$ (۳-۱, ۳-۲, ۳-۲-۴, ۳-۲-۵, ۳-۲-۵-۶)

۱ + ۱ + ۲ + ۲ + ۳ = ۹

$C_{۴} = \frac{۵}{۱۰} = ۰/۵$ (۴-۲, ۴-۲-۱, ۴-۲-۳, ۴-۲-۵, ۴-۲-۵-۶)

۱ + ۲ + ۲ + ۲ + ۳ = ۱۰

$C_{۵} = \frac{۵}{۸} = ۰/۶۲۵$ (۵-۲-۱, ۵-۲, ۵-۲-۳, ۵-۲-۴, ۵-۶)

۲ + ۱ + ۲ + ۲ + ۱ = ۸

$C_{۶} = \frac{۵}{۱۲} = ۰/۴۱۶۶۶۶۶۷$ (۶-۵-۲-۱, ۶-۵-۲, ۶-۵-۲-۳, ۶-۵-۲-۴, ۶-۵)

۳ + ۲ + ۳ + ۳ + ۱ = ۱۲

betweenness

$C_2 = \dfrac{\text{تعداد تمام کوتاه‌ترین مسیرهایی که ۲ بینشان هست}}{\text{تعداد تمام کوتاه‌ترین مسیرهای } s \neq v \neq t}$

۱، ۲، ۳، ۵

$C_1 = C_3 = C_4 = C_6 = 0$

All shortest path

۱-۲، ۱-۳، ۱-۲-۴، ۱-۲-۵، ۱-۲-۵-۶، ۲-۳، ۲-۴، ۲-۵، ۲-۵-۶، ۳-۲-۴، ۳-۲-۵، ۳-۲-۵-۶

۴-۲-۵، ۴-۲-۵-۶، ۵-۶

$C_2 = \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} = 8$

(۱-۲-۴، ۱-۲-۵، ۱-۲-۵-۶، ۳-۲-۴، ۳-۲-۵، ۳-۲-۵-۶، ۴-۲-۵، ۴-۲-۵-۶)

$C_5 = 4 = \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} = 4$

(۱-۲-۵-۶، ۴-۲-۵-۶، ۳-۲-۵-۶، ۲-۵-۶)

(۱-۲، ۱-۳، ۱-۲-۴، ۱-۲-۵، ۱-۲-۵-۶، ۲-۳، ۲-۴، ۲-۵، ۲-۵-۶، ۳-۲-۴، ۳-۲-۵، ۳-۲-۵-۶

۴-۲-۵، ۴-۲-۵-۶، ۵-۶)

کاربر ۲ دارای بیشترین درجه با ارزش ۴ است و کاربران ۴ و ۶ دارای کمترین با مقدار ۱

کاربر ۲ دارای بیشترین closness است. کاربر ۶ دارای کمترین مقدار

کاربر ۲ دارای بیشترین Betweenness است و کاربران ۱ و ۳ و ۴ و ۶ دارای کمترین

درجه یک کاربر می‌تواند نشان‌دهنده محبوبیت یا سطح فعالیت آن کاربر در شبکه باشد. از closness می‌توان برای اندازه‌گیری اینکه کاربر با چه سرعتی می‌تواند اطلاعات یا نفوذ را از طریق شبکه منتشر کند و از Betweenness برای شناسایی کاربرانی که به عنوان واسطه‌های مهم یا پل بین بخش‌های مختلف شبکه عمل می‌کنند استفاده کرد.

با مقایسه کاربران براساس این معیارهای مرکزیت می‌توانیم رهبران و تاثیرگذاران اصلی در شبکه و همچنین کاربران غیرعادی یا قطع ارتباط بالقوه را که ممکن است به بررسی بیشتر نیاز داشته باشند شناسایی کنیم.

به عنوان مثال در شبکه اجتماعی معین کاربر ۲ بر اساس هر سه معیار مرکزیت تاثیرگذارترین کاربر است که نشان می‌دهد این کاربر احتمالا نقش بازیگر کلیدی و تاثیرگذار در شبکه است. کاربران ۴ و ۶ کمترین درجه و closeness را دارند که نشان می‌دهد ممکن است تعامل و نفوذ محدودی در شبکه داشته باشند. کاربران ۱ و ۳ و ۵ و ۶ کمترین Betweenness را دارند که نشان می‌دهد آنها واسطه‌های مهمی بین بخش‌های مختلف شبکه نیستند. به طور کلی این معیارهای مرکزیت می‌توانند بینش‌های ارزشمندی را برای مهندسی ویژگی در مدل‌های یادگیری ماشین ارائه دهند که رفتار شبکه اجتماعی یا تعامل کاربر را پیش‌بینی می‌کنند.

ب)

$$e_v = \frac{\text{تعداد یال‌ها در بین همسایگی‌های راس } v \text{ هستند}}{\binom{k_v}{2}}$$

$$e_1 = \frac{1}{\binom{2}{2}} = 1$$

$$e_2 = \frac{1}{\binom{4}{2}} = \frac{1}{6} = 0/16667$$

$$e_3 = \frac{1}{\binom{2}{2}} = 1$$

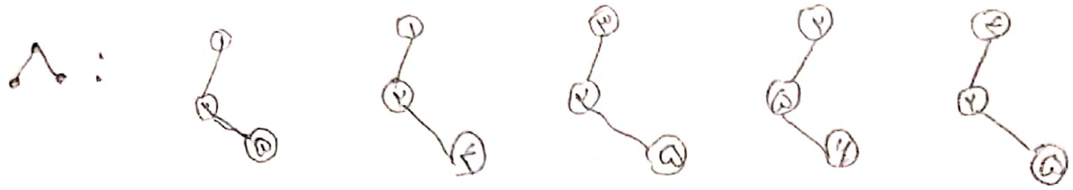
$$e_4 = e_5 = e_6 = 0$$

$K = 3$

g_1 g_2 g_3 g_4

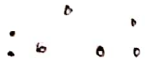
g_1 :

g_2 :



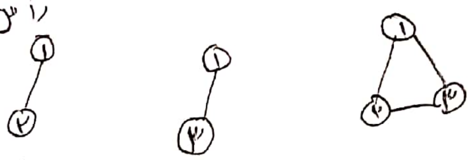
g_3 :

g_4 :



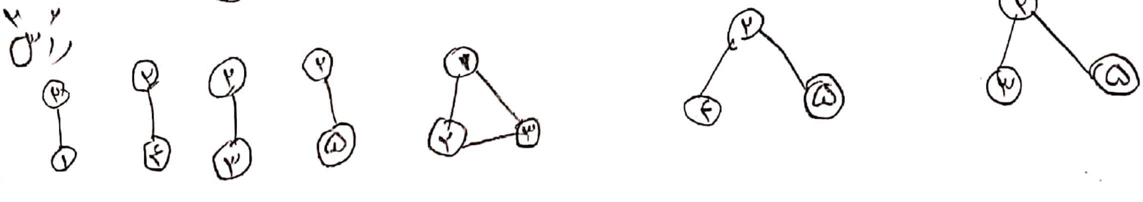
$f_\sigma = (1, ۵, ۹, ۰)^T$

راس ۱



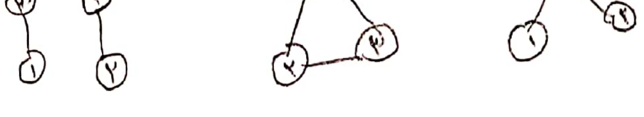
a b c d
[۲, ۱, ۰, ۰]

راس ۲



a b c d
[۴, ۱, ۲, ۰]

راس ۳



a b c D
[۲, ۱, ۱, ۰]

راس ۴

a b c d
[۱, ۰, ۰, ۰]

راس ۵

a b c d
[۲, ۰, ۱, ۰]

رأس ۶

$$\begin{array}{cccc} a & b & c & d \\ \end{array}$$
[۱، ۰، ۰، ۰]

ب) ضریب خوشه‌بندی معیاری است که نشان می‌دهد یک گراف چقدر محکم است کمیت می‌کند که رأس‌ها چقدر تمایل به تشکیل مثلث دارند. در این جا رأس ۱ و ۳ دارای ضریب خوشه‌بندی ۱ هستند به این معنی که همه همسایگان آنها به یکدیگر متصل هستند. رأس ۲ دارای ضریب خوشه‌بندی ۰/۱۶۶ است که نشان می‌دهد نسبت به رأس‌های ۱ و ۳ مثلث‌های کمتری دارد. رأس‌های ۴ و ۵ و ۶ دارای ضریب خوشه‌بندی ۰ هستند به این معنی که جزو هیچ مثلثی نیستند

گرافلت اندازه ۳ تعداد زیرگراف‌های غیرهم‌شکلی متصل با اندازه ۳ در گراف است. در اینجا رأس ۲ دارای بالاترین تعداد گرافلت ۷ است به این معنی که در بیشتر زیرگراف‌های سایز ۳ درگیر است. رأس ۵ و ۱ و ۳ به ترتیب دارای تعداد گرافلت‌های ۳، ۳ و ۴ هستند که نشان می‌دهد در تعداد کمتری درگیر هستند رأس‌های فرعی با اندازه ۳ نسبت به رأس ۲. رأس‌های ۴ و ۶ دارای تعداد گرافلت ۱ هستند به این معنی که آنها فقط در یک نوع زیرگراف با اندازه ۳ درگیر هستند

از نظر مقایسه می‌توان دید که رأس ۲ تعداد گرافلت‌های بالاتر و ضریب خوشه‌بندی کمتر است نسبت به رأس‌های ۱ و ۳. این نشان می‌دهد که رأس ۲ در زیرگراف‌های بیشتری با اندازه ۳ درگیر است اما مثلث‌های کمتری نسبت به رأس ۱ و ۳ دارد. رأس‌های ۴ و ۶ کمترین مقادیر را برای هر دو اندازه‌گیری دارد که نشان می‌دهد در مقایسه با رأس‌های دیگر در گراف کمتر به هم متصل هستند و کمتر درگیر زیرگراف‌های اندازه ۳ هستند.

ج) کاربری که Betweenness بیشتری دارد احتمالاً بیشترین تأثیر را بر تعاملات شبکه دارد. این به این دلیل است که Betweenness تعداد بارهایی که یک کاربر به عنوان پل در مسیر کوتاه‌ترین مسیر بین ۲ کاربر دیگر در شبکه عمل می‌کند را اندازه می‌گیرد. بنابراین یک کاربر با Betweenness بالا توانایی بیشتری برای کنترل جریان اطلاعات و تأثیرگذاری بر تعاملات بین دیگر کاربران در شبکه دارد. البته این فقط یک معیار است و ممکن است عوامل دیگری وجود داشته باشد که در شناسایی بهترین استفاده از شبکه‌های اجتماعی برای تبلیغات هدفمند مؤثر باشد. به عنوان مثال یک کاربر با نمره مرکزیت درجه بالا ممکن است تعداد زیادی از ارتباطات مستقیم داشته باشد و ممکن است بتوانند به مخاطبان بیشتری دسترسی پیدا کنند در حالی که یک کاربر با نمره closeness بالا در شبکه محلی خود تأثیرگذار باشد. برای شناسایی بهترین استفاده از شبکه‌های اجتماعی برای تبلیغات هدفمند تحلیل جامعی از رفتار شبکه و رفتار کاربر لازم است.

د)

Distant Based feature

$$S_{12} = S_{23} = S_{24} = S_{25} = S_{56} = 1$$

$$S_{14} = S_{15} = S_{26} = S_{34} = S_{35} = S_{45} = 2$$

$$S_{16} = S_{36} = S_{46} = 3$$

local neighborhood:

$\underbrace{\text{common neighbors}}_{cn}$, $\underbrace{\text{jaccard's coefficient}}_{JC}$, $\underbrace{\text{Adamic-Adar}}_{aa}$

$|N(1) \cap N(2)| = |\{3\}| = 1$ $\quad |N(1) \cup N(2)| = |\{3, 4, 5, 1, 2\}| = 5$

$cn = 1$

$\frac{1}{5} = 0.2 = JC = \frac{cn}{|N(1) \cup N(2)|}$ $\quad aa = \frac{1}{\log k_3} = \frac{1}{\ln 2} = 1.4427$

$|N(1) \cap N(3)| = |\{2\}| = 1$ $\quad |N(1) \cup N(3)| = |\{1, 3, 2\}|$ $\quad JC = \frac{1}{3} = 0.333$

$aa = \frac{1}{\log k_2} = \frac{1}{\ln 4} = 0.721347$

$|N(2) \cap N(3)| = |\{1\}| = 1$ $\quad |N(2) \cup N(3)| = |\{2, 3, 1, 4, 5\}| = 5$

$JC = \frac{1}{5} = 0.2$ $\quad aa = \frac{1}{\log k_1} = \frac{1}{\ln 2} = 1.4427$

$|N(2) \cap N(4)| = 0$ $\quad |N(2) \cup N(4)| = |\{2, 4, 1, 3, 5\}| = 5$

$JC = \frac{0}{5} = 0$ $\quad aa = \frac{1}{\log 0} = 0$

$|N(2) \cap N(5)| = 0 \qquad |N(2) \cup N(5)| = |\{2, 5, 6, 1, 3, 4\}| = 6$

$JC = \frac{0}{6} = 0 \qquad aa = \frac{1}{\log 0} = 0$

$|N(5) \cap N(6)| = 0 \qquad |N(5) \cup N(6)| = |\{5, 6, 2\}| = 3$

$JC = \frac{0}{3} = 0 \quad aa = \frac{1}{\log 0} = 0$

همسایگی های مشترک common neighbors: یال های بین ۲ و ۶، ۳ و ۲، ۲ و ۱ بیشترین تعداد همسایه های مشترک را دارند که اهمیتشان را نسبت به سایر یال ها بیشتر می کند

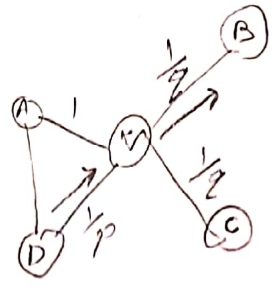
Jaccard's: یال های بین راس های ۱ و ۳ دارای بالاترین ضریب جاکارد هستند که نشان می دهد بیشترین همسایگی ها را دارند بنابراین اهمیت بیشتری دارند

Adamic-Adar: بین یال های ۱ و ۲، ۳ و ۲ دارای بالاترین شاخص Adamic-Adar است که نشان دهنده اهمیت بیشتر آن ها نسبت به سایر یال ها است

distant: یال های بین ۲ و ۶، ۳ و ۲، ۴ و ۲، ۵ و ۲، ۶ و ۲ دارای کمترین فاصله هستند که نشان می دهد به دلیل اتصال بیشتر به گره های دیگر در شبکه اهمیت بیشتری دارند

سؤال ۲: الف) در الگوریتم node2vec به ازای تمام رئوس آن شبکه یک تعداد قدم زدن تصادفی در نظر می‌گیریم طول قدم زدن تصادفی را fix می‌کنیم و به ازای آن رأس‌ها تمام رأس‌هایی که در همسایگی آن رأس هستند به ازای قدم زدن تصادفی می‌گذاریم $N_R(u)$. در این روش به تک تک رأس‌ها به نوعی حالت وزن می‌دهد و بر اساس وزن‌ها رأس‌های همسایگی را انتخاب می‌کند و از آنها random walk می‌سازد این الگوریتم انجام یک دو روش تصادفی بایاس بر روی نمودار است جایی که قدم زدن‌های تصادفی توسط پارامترهای p و q کنترل می‌شوند که درجه BFS و DFS را کنترل می‌کنند در node2vec سعی بر این است که قدم زدن تصادفی طوری انتخاب شود که هم همسایگی‌های محلی (BFS) و هم همسایگی‌های سراسری DFS تعادل بیشتری برقرار کند و هر دو را به نوعی داشته باشد

وقتی می‌خواهیم یک رأس انتخاب کنیم اینکه در گام قبل چه رأس‌هایی انتخاب کردیم مهم است و ما این را می‌دانیم در DeepWalk تمام این کارها تصادفی بود وقتی می‌خواستیم رأس (w) را انتخاب کنیم اصلاً نمی‌دانستیم که قبلاً در رأس u بوده اطلاعاتی در مورد رأس ذخیره نشده بود و ما آن را هم اصلاً اعمال نمی‌کردیم اما در 2nd order که node2vec که یکی از اینهاست در واقع اطلاعات قبلی در انتخاب‌های بعدی اعمال می‌شوند در node2vec هر دو روش BFS و DFS در انتخاب مسیر گره‌ها اعمال می‌شود اما در DeepWalk با هدف حفظ نزدیکی محلی رأس‌ها BFS در یک گراف انجام می‌دهد



$p = ۲۰۰$ $\quad q = ۱$

$$\frac{1}{p} = \frac{1}{۲۰۰} = ۰/۰۰۵$$

$$\frac{1}{q} = \frac{1}{1} = ۱$$

فعلاً از رأس D شروع می‌کنیم و می‌خواهیم یک قدم زدن تصادفی به طول ۲ طی کنیم

$$Z = \frac{1}{p} + \frac{1}{q} + \frac{1}{q} + 1 = ۳/۰۰۵$$

$\frac{1}{p} \times \frac{1}{Z} = ۰/۰۰۱۶$ احتمال برگشت از v به D $\quad$ $1 \times \frac{1}{Z} = ۰/۳۳۳$ BFS احتمال رفتن از v به A

$\frac{1}{q} \times \frac{1}{Z} = ۰/۳۳۳$ DFS احتمال رفتن از v به B q به گام بعدی رفتن

p احتمال بازگشت

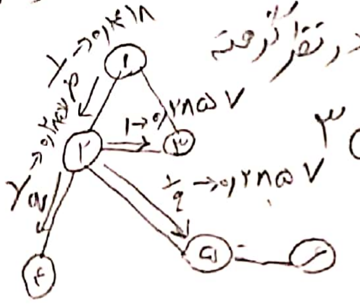
p و q را کاربر انتخاب میکند

$\frac{1}{p}$ و $\frac{1}{q}$ و ۱ احتمال های نرمال شده نیستند برای همین n را حساب کردیم و این احتمالات را در $\frac{1}{n}$ ضرب کردیم

اگر p کم باشد $\frac{1}{p}$ بزرگ است ما اهمیت بیشتری به BFS میدهیم طول ۷ و خود ۷ همسایگی محلی میشود q هم BFS را مشخص میکند چون باز میگردیم به گام اول

اگر ما مقدار p را بزرگ در نظر بگیریم بزرگتر از $\max(q,1)$ بنابراین چون $\frac{1}{p}$ در نظر میگیریم مقدار آن کوچک میشود و احتمال اینکه برگردیم به آن رأس هایی که قبلاً در آن بودیم احتمالشان کمتر میشود. اگر p کوچک باشد از $\min(q,1)$ کمتر باشد در اینصورت چون $\frac{1}{p}$ به عنوان احتمال در نظر میگیریم شخص برگردد به نقاط قبلی احتمال اینکه در همسایگی های محلی یا BFS بماند بیشتر میشود. اگر ما مقدار q را فرض کنیم بزرگتر از ۱ است بنابراین احتمال $\frac{1}{q}$ کوچک است بنابراین احتمال اینکه در DFS برویم کمتر میشود اگر $q<1$ باشد احتمال اینکه DFS انجام شود زیاد میشود. اگر $p=2$ و $q=0.5$ باشد پیاده روی تصادفی به سمت رأس های حرکت میکند که ما دور شویم هستند و اگر $p=q=2$ پیاده روی تصادفی به سمت رأس های حرکت میکند که نزدیکتر هستند که دورتر هستند

پ) یک بعد م زدن تصادفی به طول ۲ از رأس ۱ با پارامترهای $p=2$ و $q=1$ شروع میکنیم و به صورت تصادفی به رأس ۲ که رفتیم بعد از آن برای رفتن به رأس بعد میتوان به رأس ۳، ۴ و ۵ رفت یا به ۱ بازگشت برای بازگشت به رأس ۱ احتمال $\frac{1}{p}$ در نظر گرفته برای رفتن به رئوس ۴ و ۵ هر کدام احتمال $\frac{1}{q}$ و رفتن به رأس ۳ احتمال ۱ در نظر میگیریم



برای نرمالایز کردن مقادیر احتمالات داریم:

$$n = \frac{1}{p} + \frac{1}{q} + \frac{1}{q} + 1 = 0.5 + 1 + 1 + 1 = 3.5$$

$$\frac{1}{p} = \frac{1}{2} = 0.5$$

$$\frac{1}{q} = \frac{1}{1} = 1$$

$$\frac{1}{p} \times \frac{1}{n} = 0.1428$$

$$\frac{1}{q} \times \frac{1}{n} = 0.2857$$

$$1 \times \frac{1}{n} = 0.2857$$

بنابراین ۱ ≃ ۰/۲۸۵۷ + ۰/۲۸۵۷ + ۰/۲۸۵۷ + ۰/۱۴۱۸ بر اساس این احتمالات ما رفتن به راس بعدی را انتخاب می کنیم چون مقادیر $\frac{1}{p}$ و $\frac{1}{q}$ بیشتر از $\frac{1}{pq}$ هستند پس ما به راس ۱ برنمی گردیم و می توانیم یا به ۳ یا به ۴ یا به ۵ برویم برای مثال ما رفتن به ۵ را انتخاب می کنیم بنابراین راه رفتن تصادفی نمونه برداری شده به طول ۲ که از راس ۱ شروع می شود [۱و۲و۵] است

ب) node2vec: نقاط قوت: امکان جست و جوی منعطف تر گراف ها را با تنظیم پارامترهای p و q فراهم می کند که تعادل بین BFS و DFS را کنترل می کند. و اینطور نیست که فقط بر اساس احتمالات یکسان انتخاب شوند
برای حفظ ساختار گراف محلی و جهانی طراحی شده و برای کارهای پایین دستی که به هر دو نوع اطلاعات متکی هستند مفید است
این الگوریتم را می توان با استفاده از تکنیک های موازی سازی روی گراف های بزرگ آموزش داد.
پیچیدگی زمانی به طور خطی است پیچیدگی محاسباتی کم است به دلیل اینکه ما طول قدم زدن تصادفی را fix کردیم بعد از آنها را یک چیز ثابت کردیم و ما فقط با راس هایی کار می کنیم که روی قدم زدن تصادفی هستند این مراحل به قوی به هم وابسته نیستند
روشی سریع است.

معایب: به تنظیم چندین هایپرپارامتر مانند p و q و طول قدم زدن تصادفی و تعداد قدم ها نیاز دارد که می تواند زمان بر باشد.
ممکن است در گراف هایی با توزیع درجه بسیار منحرف یا ساختارهای جامعه پیچیده به خوبی عمل نکند.

DeepWalk: نقاط قوت: یک الگوریتم ساده تر است که پیاده سازی آن آسان است و می تواند برای طیف گسترده ای از گراف ها مورد استفاده قرار گیرد.
می توان با استفاده از SGD تصادفی می توان DeepWalk را آموزش داد و می تواند گراف های بزرگ با تعداد زیاد راس و یال را مدیریت کند

معایب: فقط ساختار گراف محلی را حفظ می کند و ممکن است به اندازه node2vec اطلاعات سراسری را ضبط نکند
اجازه جست و جوی انعطاف پذیر را در گراف نمی دهد.

ممکن است در گراف‌های با ساختارهای پیچیده اجتماعی یا ویژگی‌های رأس با ابعاد بالا به خوبی عمل نکند

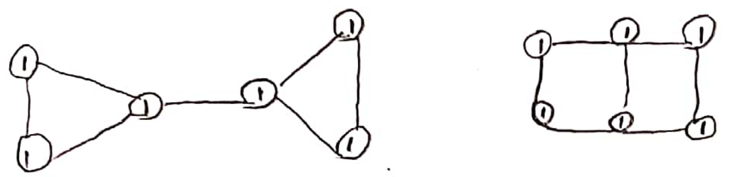
اگر گراف دارای ساختارهای جامعه پیچیده است و این کار به اطلاعات محلی و جهانی نیاز دارد node2vec ممکن است انتخاب بهتری باشد

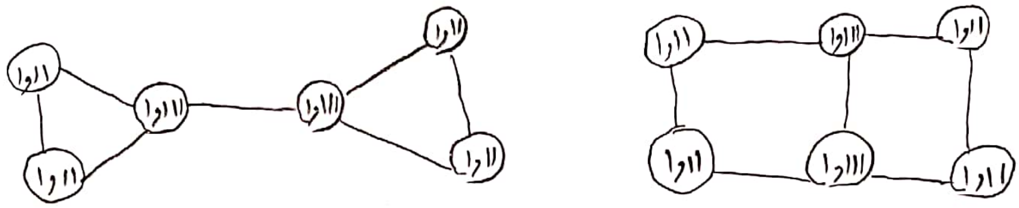
اگر مسئله ما مسئله طبقه بندی است node2vec بهتر عمل می کند

اگر گراف ساده است و فقط به اطلاعات محلی نیاز دارد DeepWalk ممکن است برای پیاده‌سازی کافی و آسان‌تر باشد.

سوال ۳:

الف)

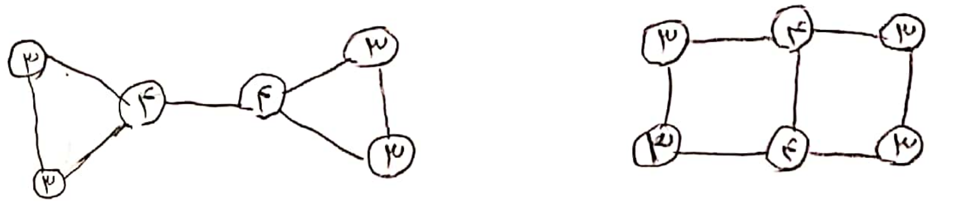


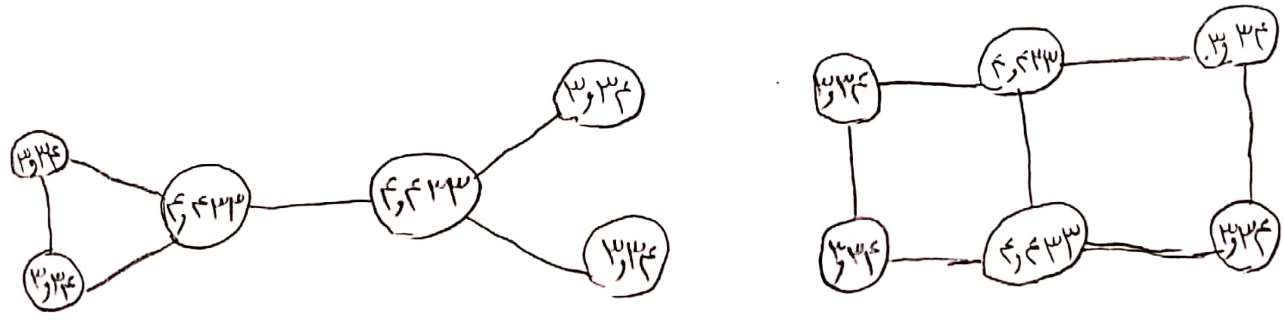


hash

۱و۱۱ → ۳

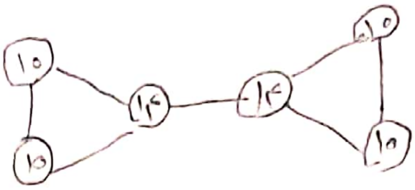
۱و۱۱۱ → ۴

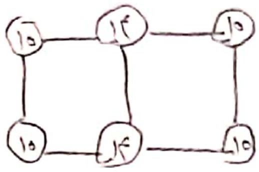




۳و۳۴ → ۱۰

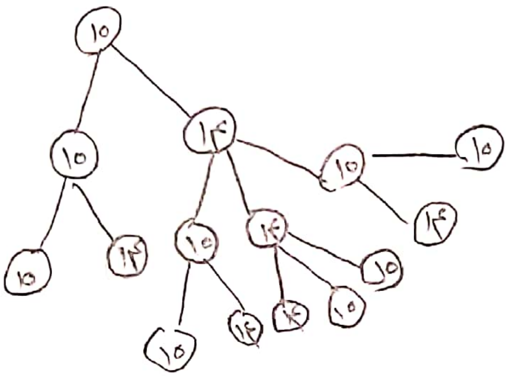
۴و۴۳۳ → ۱۴

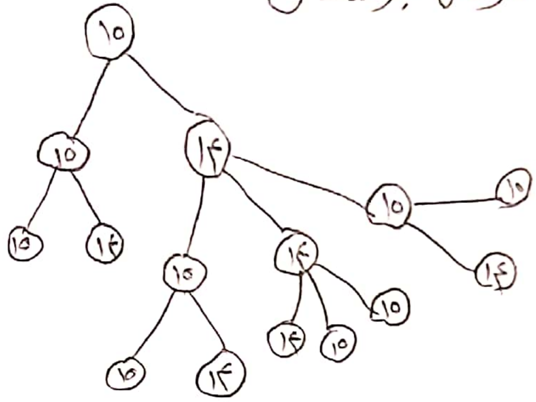




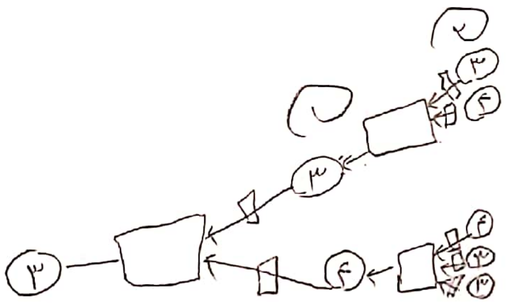
الگوریتم وایسفایلر-لیمن قادر به تشخیص عدم یک ریختی گراف بالا نیست

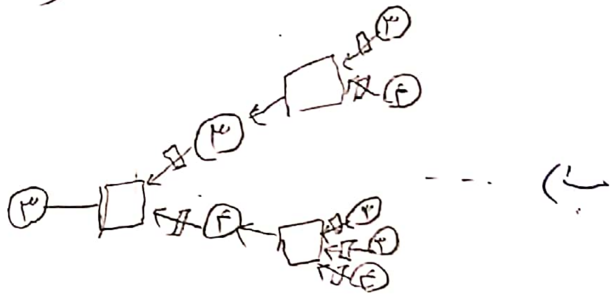
نمونه‌ای از یک گراف با تعداد درخت (unfolding tree) در حال باز شدن





هر دو گراف درخت های باز شده معادل دارند





computation Graph برای هر دو گراف تا یک مرحله‌ای رسم شده همان طور که می‌بینید یکسان است

$\text{mean}(3,4)=\frac{7}{2}$

$\text{mean}(3,4)=\frac{7}{2}$

$h^{0}=3$

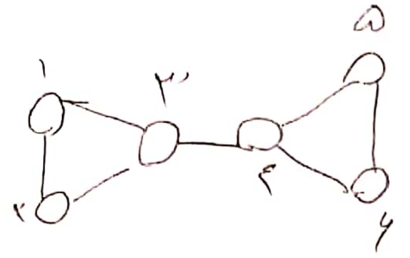
$h^{1}=\frac{7}{2}+1\cdot 3=\frac{13}{2}$

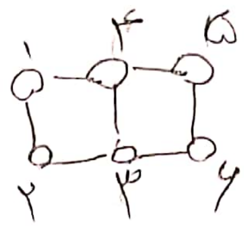
$h^{2}_{3}=\frac{7}{2}+\frac{13}{2}=\frac{20}{2}=10$

برای هر دو گراف یکسان است

پس همانطور که در قسمت الف مشخص شد هر دو گراف computationgraph یکسانی دارند.

۲۰) - ابتدا شکل ها را در نظر میگیریم



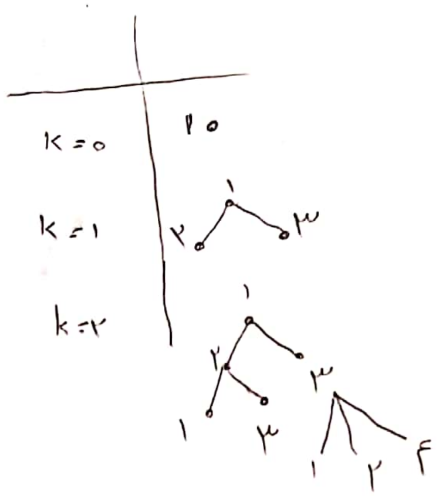


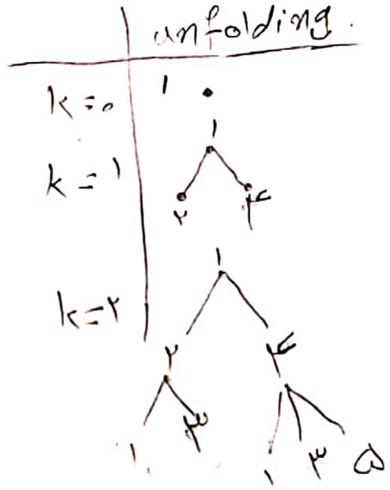
ه:

نمی‌توان بین رئوس ۲ گراف تناظر ایجاد کنیم و یک تناظر پیدا کنیم که رئوس های یک گراف را به رئوس گراف دیگر ببرد و نمی توان یال ها را حفظ کرد پس غیر یکریخت هستند

۱ → ۱   ۲ →

۳ → ۲





برای تعیین یکریختی ۲ گراف باید به چند مورد توجه شود:

تعداد رئوس: ۶

تعداد یال ها: ۷

درجه هر یک از رئوس:

همسایگی هر راس

$k_1: 2$, $k_2: 2$, $k_3: 3$, $k_4: 3$, $k_5: 2$, $k_6: 2$

$k_1: 2$, $k_2: 2$, $k_3: 3$, $k_4: 3$, $k_5: 2$, $k_6: 2$

$N(1) = \{2, 3\}$, $N(2) = \{1, 3\}$, $N(3) = \{1, 2, 4\}$, $N(4) = \{3, 5, 6\}$, $N(5) = \{4, 6\}$, $N(6) = \{4, 5\}$

$N(1) = \{2, 4\}$, $N(2) = \{1, 3\}$, $N(3) = \{2, 4, 6\}$, $N(4) = \{1, 3, 5\}$, $N(5) = \{4, 6\}$, $N(6) = \{3, 5\}$

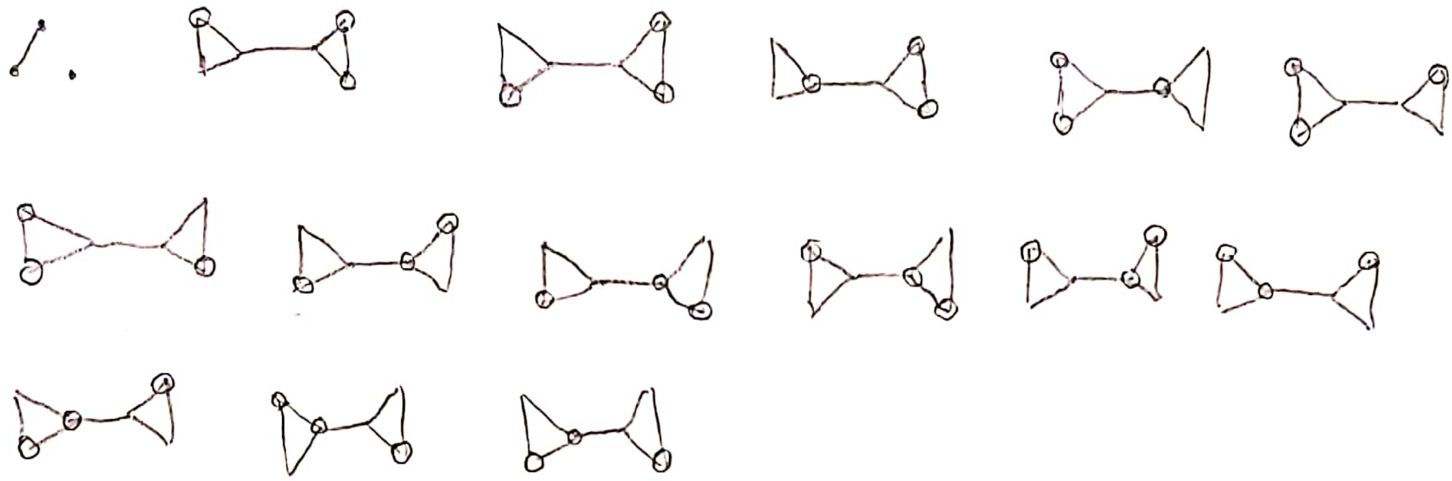
شمارش گرافلت ها:

k = ۳

$g_{۱}$

$g_{۲}$

$g_{۳}$



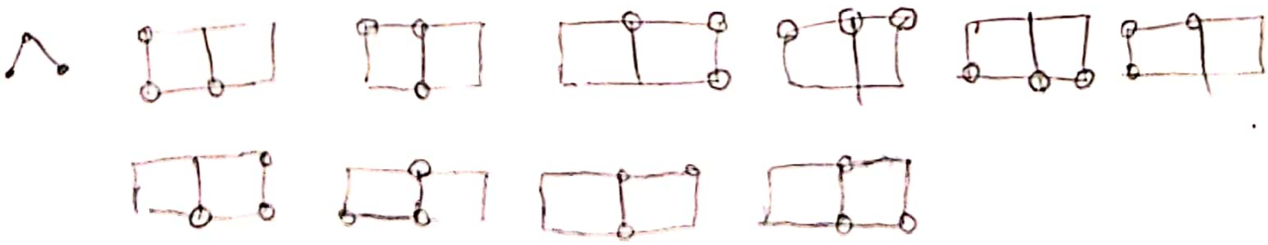
$g_{۴}$

$f_{G_{۱}} = (۲, ۴, ۱۴, ۰)^T$

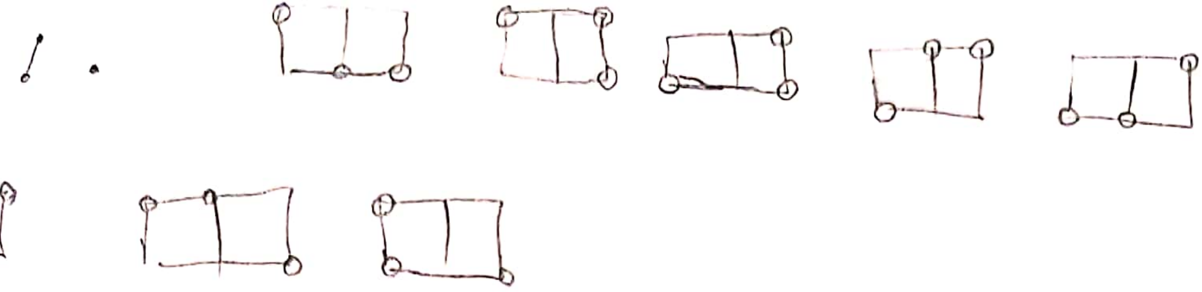
$r = ۳$

$h_{۱}$:

$h_{۲}$:



h_3 :



h_4 : . .

$$f_H = (0,10,9,0)^T$$

یکی از راه های تشخیص غیر یکنواخت بودن دو گراف استفاده از Graphlet ها است ما در اینجا Graphlet هایی که به اندازه ۳ برای هر دو گراف در نظر گرفتیم همان طور که مشاهده می شود $f_{G_1} \neq f_H$ است پس دو گراف یکنواخت نیستند

سوال امتیازی:

Expressivity: Graphlet و WL روش های مبتنی بر گراف بسیار رسا هستند به این معنی که می توانند اطلاعات دقیقی در مورد ساختار گراف بگیرند به عنوان مثال WL می تواند بین زیرگراف های هم شکل تمایز قائل شود در حالی که GCN ها نمی توانند از طرف دیگر GCN ها نسبت به Graphlet و WL بیان کمتری دارند اما مقیاس پذیرتر هستند و می توانند گراف های بزرگتری مدیریت کنند

استحکام در آشفتگی های گراف: Graphlet و WL برای آشفتگی های گراف مانند اضافه کردن یا حذف کردن رأس ها و یال ها بسیار قوی هستند یعنی representation های آموخته شده از گراف پایدار هستند حتی زمانی که گراف کمی تغییر می کند در مقابل GCN ها به اختلالات گراف حساس ترند و تغییرات کوچک در گراف می توانند تا حد زیادی بر representation های آموخته شده داشته باشند

کارایی داده: GCN ها معمولاً نسبت به Graphlet و WL از نظر داده کارآمدتر هستند یعنی می توانند representation های خوبی را با داده های کمتر یاد بگیرند زیرا که GCN ها می توانند ساختار گراف و ویژگی رئوس را به طور همزمان یاد بگیرند در حالی که Graphlet و WL تنها به ساختار گراف متکی هستند با این حال نقاط قوت GCN ها می تواند نیز ضعف باشد زیرا ممکن است در یادگیری ساختارهای گراف به تنهایی بدون ویژگی های گره موثر نباشند.

کارایی محاسباتی: GCN می تواند از نظر محاسباتی هزینه زیادی داشته باشد به خصوص برای گراف های بزرگ زیرا شامل عملیات ماتریسی می شود WL به طور کلی سریع تر از GCN است اما به دلیل تکرارهای متعدد برچسب گذاری و برچسب گذاری مجدد هنوز

مقداری سربار محاسباتی دارد. از طرف دیگر Graphlet از نظر محاسباتی کارآمد است و می‌تواند برای گراف‌های بزرگ استفاده شود.

الزامات داده: GCN به مقادیر زیادی از داده‌های آموزشی برچسب‌گذاری شده برای عملکرد بهینه نیاز دارد که می‌تواند برای برخی برنامه‌ها چالش‌برانگیز باشد. WL به مقداری داده برچسب‌دار برای مرحله برچسب‌گذاری اولیه نیاز دارد اما این می‌تواند کمتر از آنچه GCN لازم دارد باشد.

Graphlet به داده‌های برچسب‌گذاری شده نیازی ندارد بنابراین برای کارهای یادگیری بدون نظارت مناسب‌تر است

GCN برای ساختارهای گراف پیچیده با مقادیر زیادی از داده‌های برچسب‌گذاری شده مناسب است WL برای گراف‌های کوچک‌تر یا آنهایی که ساختارهای واضح دارند مناسب‌تر و Graphlet برای کارهای یادگیری بدون نظارت و گراف‌های بزرگ با ساختار محلی مناسب‌تر است.