

به نام خدا



دانشگاه صنعتی شاهرود  
Shahrood University of Technology

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

عنوان تمرین : تحلیل شبکه های اجتماعی با استفاده از مدل های پوچ

استاد: دکتر محسن رضوانی

نویسنده: مریم درویشیان

تاریخ: 1402/02/07

## دستورالعمل ها :

جمع آوری مجموعه داده: مجموعه داده یک شبکه اجتماعی را پیدا کنید .می توانید از در دسترس عموم استفاده کنید منابعی مانند اسنپ (پروژه تحلیل شبکه استنفورد) .  
شرح مختصری از مجموعه داده مورد استفاده در تکلیف ارائه دهید.

می‌توانیم از مجموعه داده شبکه اجتماعی فیسبوک که به صورت عمومی در دسترس استفاده قرار می‌گیرد استفاده کنیم. این مجموعه داده شامل گراف اندازه گیری شده از شبکه ارتباطی بین کاربران فیسبوک است و دارای بیش از چهار میلیارد یال بین دو نفر از کاربران فیسبوک است. این مجموعه داده از ۲۲۳ میلیون کاربر و ۲۷۲ میلیون یال شبکه تشکیل شده است. برای دسترسی به این مجموعه داده لینک زیر را استفاده کردم:

<https://snap.stanford.edu/data/egonets-Facebook>

```
import pandas as pd

url='https://snap.stanford.edu/data/facebook_combined.txt.gz'

df=pd.read_csv(url, delimiter=' ', header=None, names=['from', 'to'])

print(df)
```

✓ 3.1s Python

	from	to
0	0	1
1	0	2
2	0	3
3	0	4
4	0	5
...	...	...
88229	4026	4030
88230	4027	4031
88231	4027	4032
88232	4027	4038
88233	4031	4038

[88234 rows x 2 columns]

nodeId.edges:

یال‌های موجود در شبکه یک فرد خاص را نشان می‌دهد. یال‌ها برای شبکه اجتماعی فیسبوک بدون جهت هستند و برای شبکه‌های توییتر دارای جهت هستند(که نماینده ی این است که شخص اول فرد دوم را فالو می‌کند). فرد خاص در این فایل لیست نشان داده نمی‌شود، اما فرض می‌شود که هر نودی که در فایل حضور دارد، فالوور فرد خاص است.

nodeId.circles:

مجموعه دایره‌ها برای فرد خاص (نود ایگو) را نمایش می‌دهد. هر خط شامل یک دایره است که شامل یک سری آی دی یا نودها اولین مقدار هر خط، نام دایره است .می باشد.

nodeId.feat:

ویژگی های هر یک از نودهایی که در فایل یال قید شده است را نمایش می دهد.

nodeId.egofeat:

ویژگی های فرد ایگو (شخص خاصی که برایش فایل ساخته می شود) نمایش داده می شود.

nodeId.featnames:

نام ویژگی های موجود در نودها نمایش داده می شود. در صورتی که یک کاربر خاص خصاییتی در پروفایلش داشته باشد، مقدار یک در نظر گرفته شده است و در غیر اینصورت، مقدار صفر در نظر گرفته می شود.

nodeId.egofeat:

ویژگی های فرد ایگو (شخص خاصی که برایش فایل ساخته می شود) نمایش داده می شود.

nodeId.featnames:

نام ویژگی های موجود در نودها نمایش داده می شود. در صورتی که یک کاربر خاص خصاییتی در پروفایلش داشته باشد، مقدار یک در نظر گرفته شده است و در غیر اینصورت، مقدار صفر در نظر گرفته می شود.

تجزیه و تحلیل شبکه ها:

از معیارهایی مانند میانگین درجه، ضریب خوشه بندی، استفاده کنید. کوتاه ترین مسیر، معیارهای مرکزیت و غیره برای تجزیه و تحلیل شبکه ها

```
import networkx as nx
G = nx.from_pandas_edgelist(df, source='from', target='to')
print(G)
```

Python

Graph with 4039 nodes and 88234 edges





شبیه سازی داده های نمودار برای 2 یا 5 سال آینده: بر اساس مدل های تهی که خواننده اید شبیه سازی شبکه ها با ویژگی های مشابه شبکه های دنیای واقعی، بهترین مدل تهی باید انتخاب شود. دلایل بهترین انتخاب خود را بنویسید.

برای شبیه سازی گراف فیسبوک، یکی از روش های استفاده از شبکه های پیچیده است. گراف باراباسی یکی از شبکه های پیچیده است که از آن برای شبیه سازی گراف فیسبوک استفاده می شود. این شبکه با توجه به ویژگی هایی مانند تعداد گره ها و روابط، برای شبیه سازی شبکه های پیچیده استفاده می شود.

استفاده از گراف باراباسی برای شبیه سازی گراف فیسبوک به دلیل اینکه گراف فیسبوک یک شبکه پیچیده است و شامل میلیاردها کاربر و صفحه مختلف است، کار پیچیده ای است. به علاوه، تعیین اینکه در آینده چه تعداد کاربر و صفحه جدید به گراف اضافه خواهند شد، بسیار دشوار است. با استفاده از گراف باراباسی، می توانیم با تغییر پارامترهایی مانند تعداد گره ها و روابط، شبکه های پیچیده و بزرگی مانند گراف فیسبوک را به خوبی شبیه سازی کنیم و درک بهتری از رفتار شبکه به دست آوریم.

شبیه سازی برای 2 سال آینده با استفاده از گراف باراباسی:

```
# شبکه سازی برای 3 سال آینده

data = pd.read_csv('https://snap.stanford.edu/data/facebook_combined.txt.gz', sep=' ', header=None, names=['source', 'target'])

graph = nx.from_pandas_edgelist(data, 'source', 'target')

for i in range(3):
    graph = nx.barabasi_albert_graph(n=len(graph), m=3)

pos = nx.spring_layout(graph)
nx.draw(graph, pos)
plt.show()

avg_degree = sum(dict(graph.degree()).values()) / len(graph)
print('میانگین درجات:', avg_degree)

clustering_coefficient = nx.average_clustering_coefficient(graph)
print('ضریب خوشه بندی:', clustering_coefficient)

largest_cc = max(nx.connected_components(graph), key=len)
print('بزرگترین اجزای مرتبط:', largest_cc)

eigenvector_centrality = nx.eigenvector_centrality_numpy(graph)
print('مرکزیت ویژه:', eigenvector_centrality)
```









43, 7603671700916 : میانگین درجات :  
 0,03763911905490862 : حریف قوه پندو  
 {0,0083,4037,4036,4035,4034,4033,4032,4031,4030,4029,4028,4027,4026,4025,4024,4023,4022,4021,4020,4019,4018,4017,4016,4015,4014,4013,4012,4011,4010,4009,4008,4007,4006,4005,4004,4003,4002,4001,4000,3999,3998,3997,3996,3995,3994,3993,3992,3991,3990,3989,3988,3987,3986,3985,3984,3983,3982,3981,3980,3979,3978,3977,3976,3975,3974,3973,3972,3971,3970,3969,3968,3967,3966,3965,3964,3963,3962,3961,3960,3959,3958,3957,3956,3955,3954,3953,3952,3951,3950,3949,3948,3947,3946,3945,3944,3943,3942,3941,3940,3939,3938,3937,3936,3935,3934,3933,3932,3931,3930,3929,3928,3927,3926,3925,3924,3923,3922,3921,3920,3919,3918,3917,3916,3915,3914,3913,3912,3911,3910,3909,3908,3907,3906,3905,3904,3903,3902,3901,3900,3899,3898,3897,3896,3895,3894,3893,3892,3891,3890,3889,3888,3887,3886,3885,3884,3883,3882,3881,3880,3879,3878,3877,3876,3875,3874,3873,3872,3871,3870,3869,3868,3867,3866,3865,3864,3863,3862,3861,3860,3859,3858,3857,3856,3855,3854,3853,3852,3851,3850,3849,3848,3847,3846,3845,3844,3843,3842,3841,3840,3839,3838,3837,3836,3835,3834,3833,3832,3831,3830,3829,3828,3827,3826,3825,3824,3823,3822,3821,3820,3819,3818,3817,3816,3815,3814,3813,3812,3811,3810,3809,3808,3807,3806,3805,3804,3803,3802,3801,3800,3799,3798,3797,3796,3795,3794,3793,3792,3791,3790,3789,3788,3787,3786,3785,3784,3783,3782,3781,3780,3779,3778,3777,3776,3775,3774,3773,3772,3771,3770,3769,3768,3767,3766,3765,3764,3763,3762,3761,3760,3759,3758,3757,3756,3755,3754,3753,3752,3751,3750,3749,3748,3747,3746,3745,3744,3743,3742,3741,3740,3739,3738,3737,3736,3735,3734,3733,3732,3731,3730,3729,3728,3727,3726,3725,3724,3723,3722,3721,3720,3719,3718,3717,3716,3715,3714,3713,3712,3711,3710,3709,3708,3707,3706,3705,3704,3703,3702,3701,3700,3699,3698,3697,3696,3695,3694,3693,3692,3691,3690,3689,3688,3687,3686,3685,3684,3683,3682,3681,3680,3679,3678,3677,3676,3675,3674,3673,3672,3671,3670,3669,3668,3667,3666,3665,3664,3663,3662,3661,3660,3659,3658,3657,3656,3655,3654,3653,3652,3651,3650,3649,3648,3647,3646,3645,3644,3643,3642,3641,3640,3639,3638,3637,3636,3635,3634,3633,3632,3631,3630,3629,3628,3627,3626,3625,3624,3623,3622,3621,3620,3619,3618,3617,3616,3615,3614,3613,3612,3611,3610,3609,3608,3607,3606,3605,3604,3603,3602,3601,3600,3599,3598,3597,3596,3595,3594,3593,3592,3591,3590,3589,3588,3587,3586,3585,3584,3583,3582,3581,3580,3579,3578,3577,3576,3575,3574,3573,3572,3571,3570,3569,3568,3567,3566,3565,3564,3563,3562,3561,3560,3559,3558,3557,3556,3555,3554,3553,3552,3551,3550,3549,3548,3547,3546,3545,3544,3543,3542,3541,3540,3539,3538,3537,3536,3535,3534,3533,3532,3531,3530,3529,3528,3527,3526,3525,3524,3523,3522,3521,3520,3519,3518,3517,3516,3515,3514,3513,3512,3511,3510,3509,3508,3507,3506,3505,3504,3503,3502,3501,3500,3499,3498,3497,3496,3495,3494,3493,3492,3491,3490,3489,3488,3487,3486,3485,3484,3483,3482,3481,3480,3479,3478,3477,3476,3475,3474,3473,3472,3471,3470,3469,3468,3467,3466,3465,3464,3463,3462,3461,3460,3459,3458,3457,3456,3455,3454,3453,3452,3451,3450,3449,3448,3447,3446,3445,3444,3443,3442,3441,3440,3439,3438,3437,3436,3435,3434,3433,3432,3431,3430,3429,3428,3427,3426,3425,3424,3423,3422,3421,3420,3419,3418,3417,3416,3415,3414,3413,3412,3411,3410,3409,3408,3407,3406,3405,3404,3403,3402,3401,3400,3399,3398,3397,3396,3395,3394,3393,3392,3391,3390,3389,3388,3387,3386,3385,3384,3383,3382,3381,3380,3379,3378,3377,3376,3375,3374,3373,3372,3371,3370,3369,3368,3367,3366,3365,3364,3363,3362,3361,3360,3359,3358,3357,3356,3355,3354,3353,3352,3351,3350,3349,3348,3347,3346,3345,3344,3343,3342,3341,3340,3339,3338,3337,3336,3335,3334,3333,3332,3331,3330,3329,3328,3327,3326,3325,3324,3323,3322,3321,3320,3319,3318,3317,3316,3315,3314,3313,3312,3311,3310,3309,3308,3307,3306,3305,3304,3303,3302,3301,3300,3299,3298,3297,3296,3295,3294,3293,3292,3291,3290,3289,3288,3287,3286,3285,3284,3283,3282,3281,3280,3279,3278,3277,3276,3275,3274,3273,3272,3271,3270,3269,3268,3267,3266,3265,3264,3263,3262,3261,3260,3259,3258,3257,3256,3255,3254,3253,3252,3251,3250,3249,3248,3247,3246,3245,3244,3243,3242,3241,3240,3239,3238,3237,3236,3235,3234,3233,3

شبکه ها را مقایسه کنید و شباهت ها و تفاوت های آنها را شناسایی کنید.

```
import networkx as nx
import random

n = 4039
m = 1
p = 0.5

G = nx.barabasi_albert_graph(n, m, seed=None)

for node in G.nodes():
    for neighbor in G.nodes():
        if node < neighbor and not G.has_edge(node, neighbor):
            if random.random() < p:
                G.add_edge(node, neighbor)

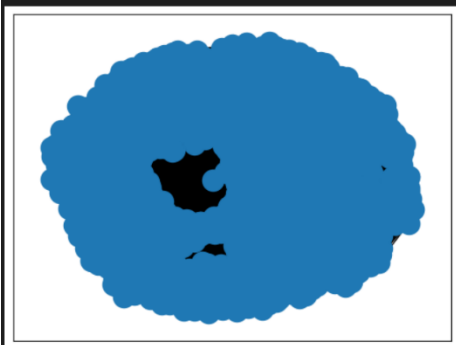
print("Number of nodes:", G.number_of_nodes())
print("Number of edges:", G.number_of_edges())

pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos)
nx.draw_networkx_edges(G, pos)
plt.show()
```

2m 36.4s

## Python

```
Number of nodes: 4039
Number of edges: 4078416
```



```

import networkx as nx
import random

n = 4039
m = 1
p = 1

G = nx.barabasi_albert_graph(n, m, seed=None)

for node in G.nodes():
    for neighbor in G.nodes():
        if node < neighbor and not G.has_edge(node, neighbor):
            if random.random() < p:
                G.add_edge(node, neighbor)

print("Number of nodes:", G.number_of_nodes())
print("Number of edges:", G.number_of_edges())

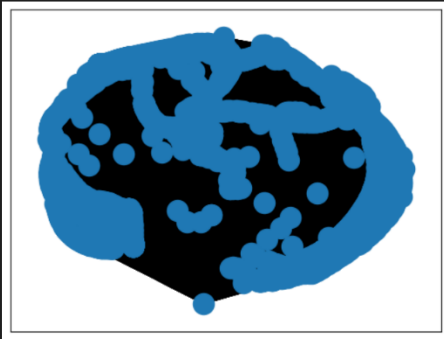
pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos)
nx.draw_networkx_edges(G, pos)
plt.show()

```

7m 56.2s

Python

Number of nodes: 4039  
Number of edges: 8154741



الگوریتم Random Walk را پیاده سازی کنید و کاربرد آن را در شبکه ایجاد شده نشان دهید با چند مثال.

```
#random walk algorithm on a barabasi albert graph
import networkx as nx
import random

n = 4039
m = 2
G = nx.barabasi_albert_graph(n, m)

num_steps = 1000
start_node = random.choice(list(G.nodes()))

current_node = start_node
visited_nodes = [current_node]
for i in range(num_steps):
    neighbors = list(G.neighbors(current_node))
    if len(neighbors) == 0:
        break
    current_node = random.choice(neighbors)
    visited_nodes.append(current_node)

print("Visited nodes:", visited_nodes)
```

Visited nodes: [2130, 1267, 1467, 1267, 653, 121, 301, 78, 1739, 78, 165, 3750, 165, 711, 165, 3750, 2391, 2858, 514, 455, 229, 10, 28, 323, 385, 8, 156, 2434, 1038, 1583, 128, 580, 32]

الگوریتم Random Walk در بسیاری از حوزه‌های علمی و صنعتی کاربرد دارد. در زیر به چند مثال از کاربردهای این الگوریتم اشاره می‌کنیم:

1. محاسبه پیامدهای حرکت داده در بازار سرمایه: این الگوریتم برای محاسبه تغییرات نوسانات بازار سرمایه و پیش‌بینی رفتار بازار استفاده می‌شود. به عنوان مثال، در حالتی که یک سهم در بازار به صورت تصادفی حرکت می‌کند، با استفاده از الگوریتم Random Walk، می‌توانیم تحلیلی از رفتار سهم در آینده داشته باشیم.
2. پیش‌بینی پراکندگی ویروس‌ها: این الگوریتم برای مدل کردن حرکت ویروس‌ها و بیماری‌ها در جامعه مورد استفاده قرار می‌گیرد. با توجه به اینکه حرکت ویروس‌ها نیز به صورت تصادفی اتفاق می‌افتد، الگوریتم Random Walk می‌تواند به پیش‌بینی پراکندگی ویروس‌ها در جامعه کمک کند.
3. بررسی رفتار کاربران در شبکه‌های اجتماعی: الگوریتم Random Walk به خوبی برای بررسی رفتار کاربران در شبکه‌های اجتماعی مانند فیس‌بوک و توییتر قابل استفاده است. با انجام Random Walk در شبکه، می‌توانیم به بررسی الگوهای رفتاری کاربران، تشکیل جمعاعات و گروه‌های مختلف و مرکزیت گره‌ها بپردازیم.
4. بررسی رفتار ذرات ماده در فیزیک: الگوریتم Random Walk برای مدل کردن حرکت ذرات ماده در یک محیط مورد استفاده قرار می‌گیرد. با انجام این الگوریتم برای مدل کردن حرکت ذرات، می‌توان استفاده کرد.

نتیجه گیری: نتیجه ای بنویسید که یافته های خود را خلاصه کنید . در مورد مزایا و معایب استفاده از هر مدل برای تحلیل شبکه های اجتماعی پتانسیل را شناسایی کنید کاربرد این مدل ها در تحلیل شبکه های اجتماعی دنیای واقعی

با توجه به ماهیت پیچیده شبکه های اجتماعی استفاده از گراف با مدل البرت باراباسی مناسب است. با احتمالهای مختلف گراف انرا و مجموع نودها و یالها رسم کردم. با استفاده از الگوریتم رندم واک میتوان توزیع یا شیوع یک پارامتر را در گراف فیس بوک یا دیگر شبکه های اجتماعی و گرافهای کاملا متصل بررسی کرد. شبیه سازی شیوع یا انتشار یا توزیع نیز قابل پیاده سازی است. مثلا میتوان میزان شیوع بیماری کرونا را در یک گراف با رندم واک پیاده سازی کرد و از انجا که کامل است همیشه به جواب میرسد.