



# Math & Systems HW4

*Teacher: Dr. Ebrahimi*

Mahdi Sadeghi  
830598035

## سوال ۱:

جهت شبیه‌سازی Random Walk ابتدا یک عدد تصادفی بین ۱ و منفی ۱ انتخاب و به موقعیت قبلی walker اضافه میشود. پس از این کار را ۲۰ بار تکرار کردیم، موقعیت آخر به توان دو را به یک لیست اضافه میکنیم. پس از آن میانگین تمام مربعات جابجایی تا آن لحظه در یک لیست دیگر (transition\_square\_means) اضافه میشود. در پایان آخرین ورودی در transition\_square\_means به عنوان ضریب انتشار شبیه سازی محاسبه میشود.

```
def question1():
    all_transitions = []
    all_transitions_squares = []
    transition_square_means = []

    # do random walk TOTAL_MOVES times
    for i in range(TOTAL_MOVES):
        x = 0

        # the random walk algorithm
        for j in range(T_PER_MOVE):
            dx = random.choice([-1, 1])
            x += dx

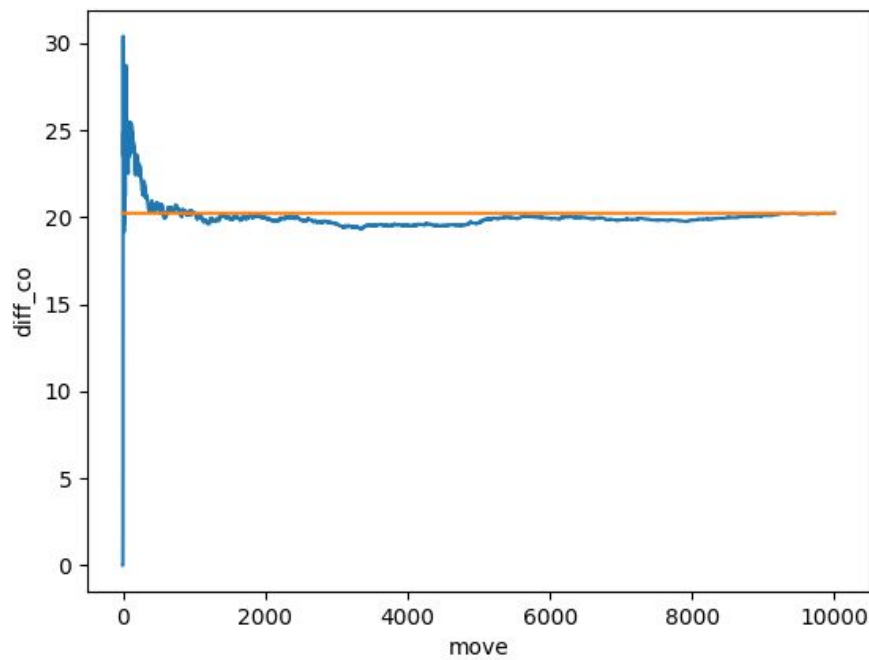
        all_transitions.append(x)
        all_transitions_squares.append(x ** 2)

    # transition square means

    transition_square_means.append(np.mean(np.array(all_transitions_squares)))

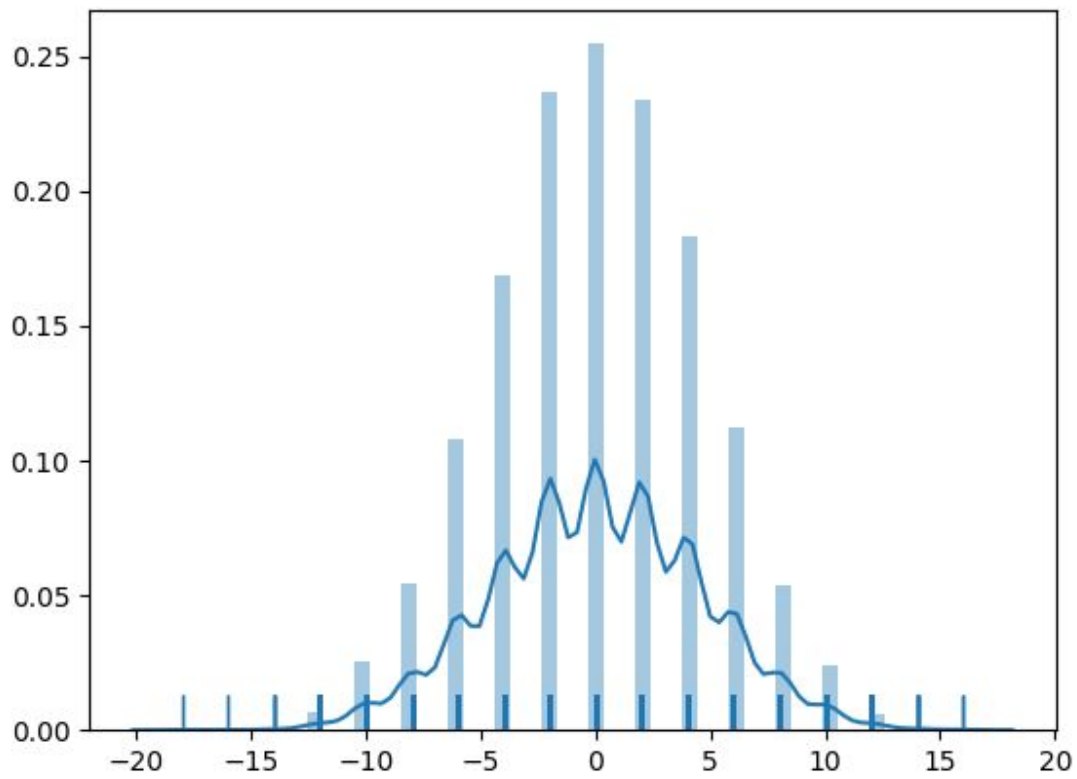
    # diffusion coefficient is equal to the last transition square
    mean we computed
    diffusion_coefficient = transition_square_means[-1]
    print('Diffusion Coefficient: {0}'.format(diffusion_coefficient))
```

ضریب انتشار به دست آمده برای ۱۰۰۰۰ بار تکرار الگوریتم بالا برابر  $20/2316$  است (البته در هر بار اجرا به دلیل احتمالی بودن حرکت Walker ضریبی متفاوت به دست می‌آید اما همگی در نزدیکی عدد ۲۰ هستند). این ضریب با تئوری سازگاری دارد، زیرا طبق تئوری امید ریاضی مربعات جابجایی برابر  $N\delta^2$  است که در اینجا برابر ۲۰ میشود.



نمودار میانگین مربعات جابجایی در سوال ۱

همچنین نمودار توزیع نیز از تئوری پیروی میکند. این نمودار به صورت نرمال با میانگین صفر است که با پیش بینی ها مطابقت دارد.



نمودار توزیع مکانی قدم زن در  $t=20$  در سوال ۱

## سوال ۲:

در سوال ۲ مشابه سوال ۱ عمل شده با این تفاوت که هنگامی که موقعیت قدم زن به ۱- میرسد، ما موقعیت را به صفر تغییر می دهیم.

```
def question2():
    all_transitions = []
    all_transitions_squares = []
    transition_square_means = []

    # do random walk TOTAL_MOVES times
    for i in range(TOTAL_MOVES):
        x = 0

        # the random walk algorithm
        for j in range(T_PER_MOVE):
            dx = random.choice([-1, 1])
            x += dx

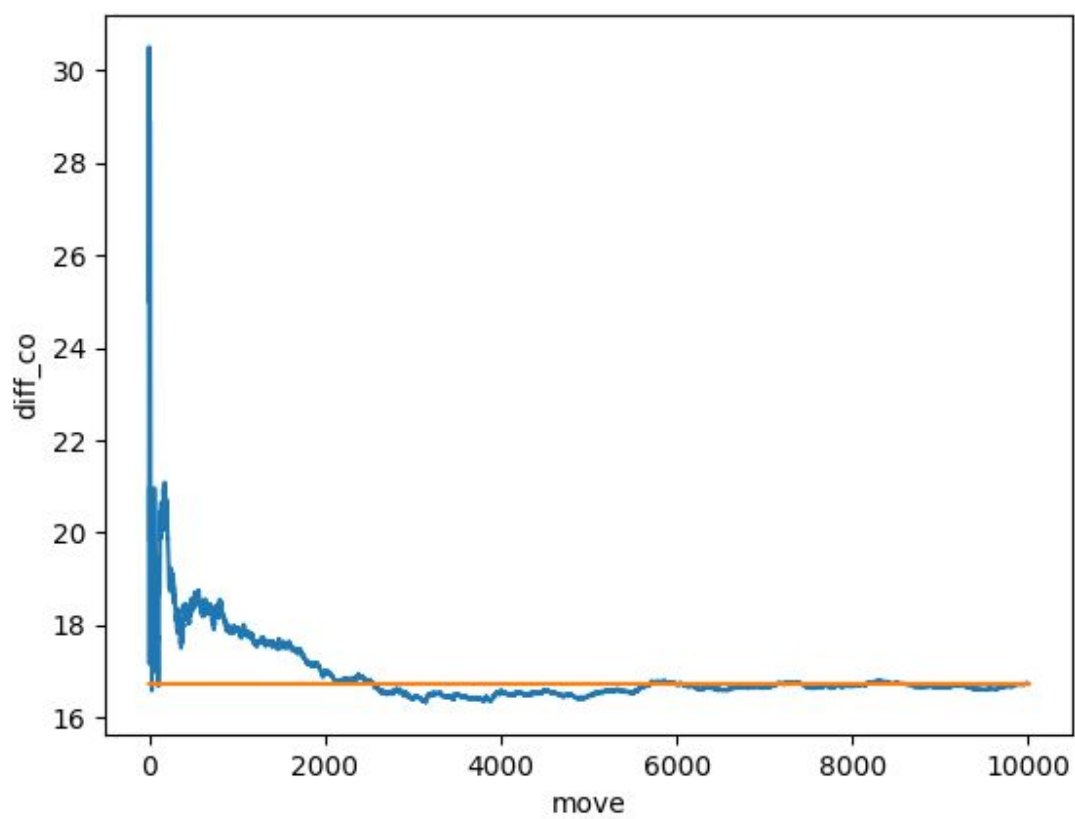
            if x == -1:
                x = 0

        all_transitions.append(x)
        all_transitions_squares.append(x ** 2)

    # transition square means
    transition_square_means.append(np.mean(np.array(all_transitions_squares)))

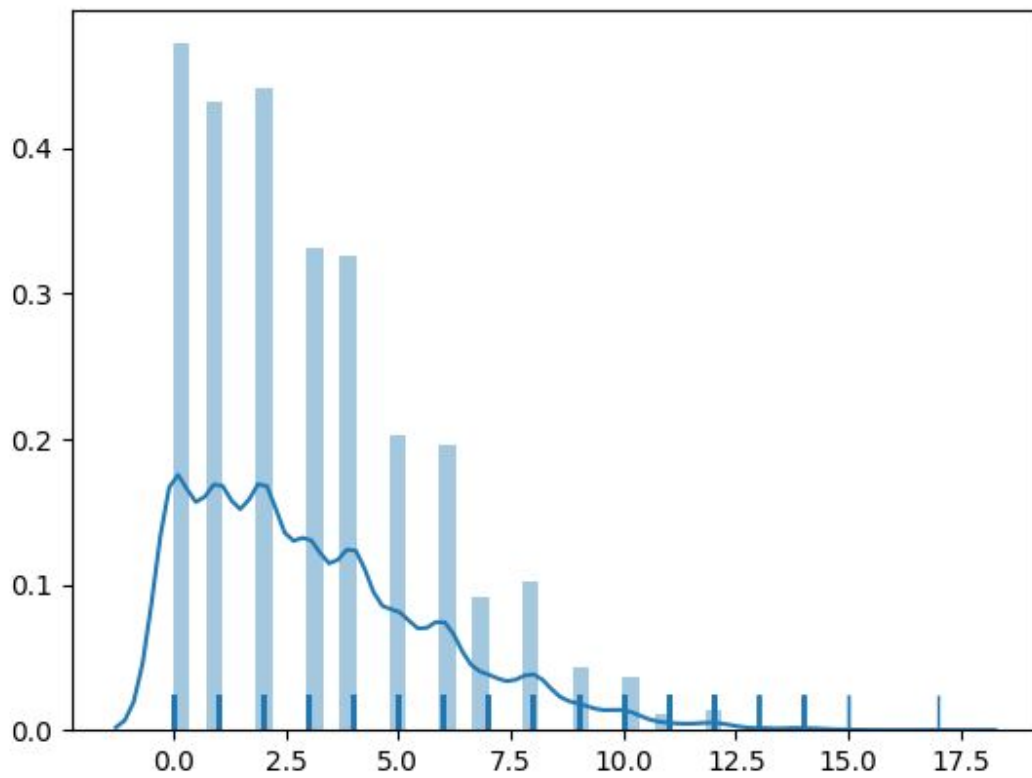
    # diffusion coefficient is equal to the last transition square
    mean we computed
    diffusion_coefficient = transition_square_means[-1]
    print('Diffusion Coefficient: {0}'.format(diffusion_coefficient))
```

ضریب انتشار به دست آمده در این حالت برابر ۱۶/۷۳۴۷ است که با تئوری (۲۰) مطابقت ندارد.



نمودار میانگین جابجایی قدم زن در سوال ۲

همچنین نمودار توزیع مکانی هم با تئوری همخوانی ندارد و چولگی بیشتری نسبت به نمودار نرمال مورد انتظار دارد.



نمودار توزیع موقعیت مکانی در  $t=20$  در سوال ۲

## سوال ۳:

در این سوال نیز همانند سوال‌های پیش عمل میکنیم با این تفاوت که هنگام حرکت کردن از یک گره به گره‌ی دیگر بجای تنها دو حالت، ما با احتمال یکسان به یکی از همسایگان گره‌ی کنونی می‌رویم. ساخت و مدیریت گراف‌ها با استفاده از کتابخانه‌ی `networkx` انجام می‌شود. تفاوت دیگری که این سوال با سوال قبل دارد این است که هنگام ساخت مربع جابجایی‌ها، از فاصله‌ی گره تا مبدا استفاده می‌کنیم:

```
for i in range(Q3_TOTAL_MOVES):
    x = 0

    # random walk algorithm
    for j in range(Q3_T_PER_MOVE):
        neighbors = list(graph.neighbors(x))
        x = random.choice(neighbors)

    # distance from origin
    x_distance = nx.shortest_path_length(graph, x, 0)

    all_transitions.append(x_distance)
    all_transitions_squares.append(x_distance ** 2)

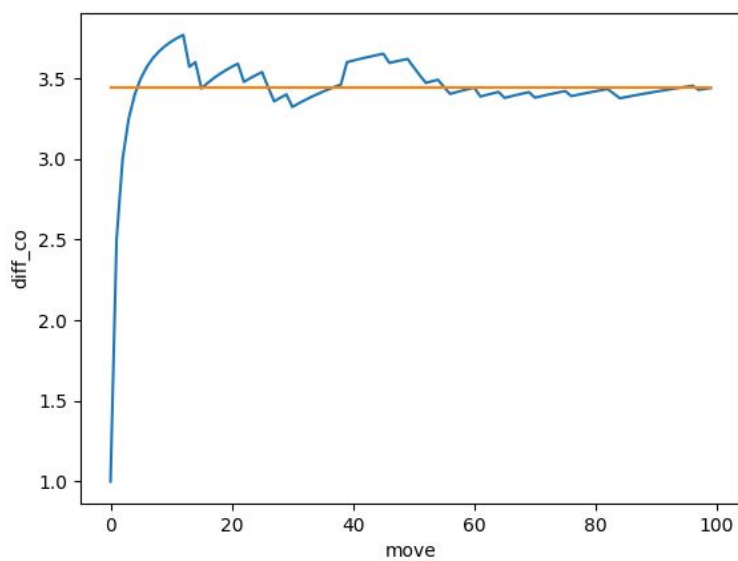
    # transition square means

transition_square_means.append(np.mean(np.array(all_transitions_squares)))

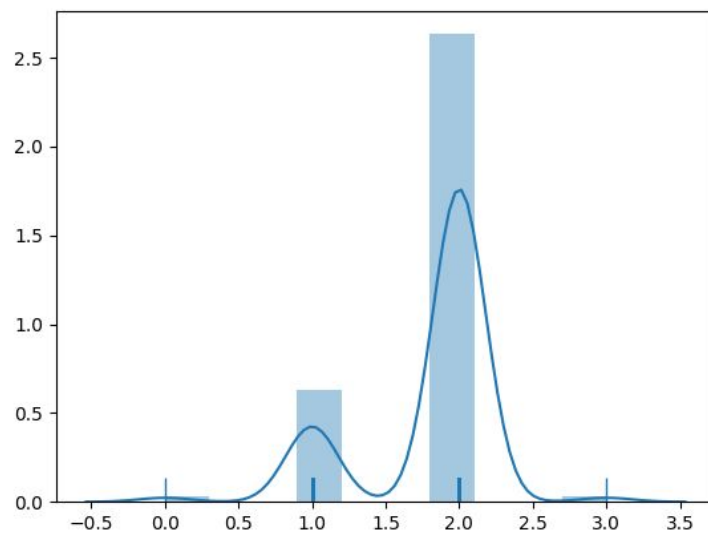
# diffusion coefficient is equal to the last transition square mean
we computed
diffusion_coefficient = transition_square_means[-1]
print('Diffusion Coefficient for {0} Network:
{1}'.format(network_type_str, diffusion_coefficient))
```

ضریب انتشار برای شبکه‌ی Erdos-Renyi برابر ۳/۴۳ بدست آمد.



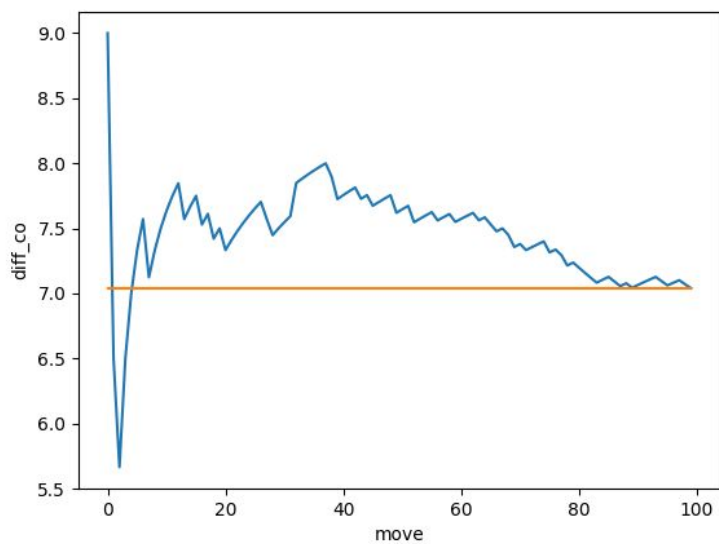


نمودار مربع جابجایی‌ها برای شبکه‌ی Erdos-Renyi

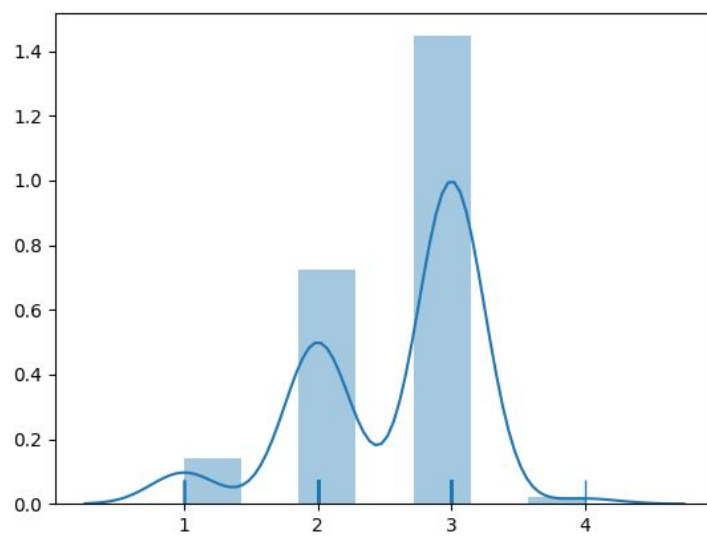


نمودار توزیع فاصله از مرکز شبکه‌ی Erdos-Renyi

ضریب انتشار برای شبکه‌ی Watts-Strogatz برابر  $7/04$  است.



نمودار مربع جابجایی‌ها برای شبکه‌ی Watts-Strogatz



نمودار توزیع فاصله از مرکز شبکه‌ی Erdos-Renyi