تمرین سری 7

علم شبکه

هانیه حاتمی 99100614

-سوال 2 فصل 6 کتاب)

a)

اگر یکی از گره‌های موجود را(i) در نظر بگیریم و درجه‌ی آن را که تعداد بارهایی است که به رقص دعوت شده، k_i می‌نامیم. تکامل زمانی درجه‌ی گره را می‌توان با در نظر گرفتن احتمال اتصال یک گره جدید به گره‌ی از قبل موجود i بدست آورد. طبق صورت سوال می‌دانیم احتمال اتصال گره جدید به گره i برابر است با:

$$P(connecting\ to\ node\ i\ at\ time\ t+1) = \frac{\eta_i}{t\langle\eta\rangle}$$

حال درجه‌ی گره i در زمان $t+1$ که برابر $k_i(t+1)$ است برابر است با درجه‌ی گره‌ی i در زمان t که همان $k_i(t)$ است و احتمال اینکه گره‌ی i در زمان $t+1$ دعوت‌نامه‌ی رقصی از گره‌ی جدید دریافت کرده باشد، و این برابر است با:

$$k_i(t+1) = k_i(t) + P(connecting\ to\ node\ i\ at\ time\ t+1) = k_i(t) + \frac{\eta_i}{t\langle\eta\rangle}$$

این معادله تکامل زمانی برای درجه گره است که نشان می‌دهد هر گره در زمان $t+1$ چه تعداد یال(یا دعوت‌نامه‌ی رقص) داشته است.

b)

برای بدست آوردن توزیع درجه، باید این احتمال را در نظر بگیریم که یک گره در زمان معین t درجه k داشته باشد. $P(k,t)$ احتمال اینکه یک گره در زمان t درجه k داشته باشد است. احتمال اینکه یک گره در زمان $t+1$ درجه k داشته باشد با مجموع دو احتمال داده می‌شود، احتمال اینکه یک گره در زمان t درجه k-1 داشته باشد و در زمان $t+1$ یک دعوت رقص دریافت کند و احتمال اینکه یک گره در زمان t درجه k داشته باشد و در زمان $t+1$ دعوتنامه رقصی دریافت نکند. که این برابر است با:

$$P(k,t+1) = P(k-1,t) \times P(\ receives\ a\ dance\ invitation\ at\ t+1\ )$$

$$+P(k,t) \times\ P\ (does\ not\ receive\ a\ dance\ invitation\ at\ t+1)$$

که احتمال دریافت نکردن دعوت برای رقص برابر با احتمال کل که 1 است منهای احتمال دریافت کردن دعوت به رقص است، که برابر می‌شود با:

$$\Rightarrow \quad P(k,t+1) = P(k-1,t) \times \frac{\eta}{t\langle\eta\rangle} + P(k,t) \times \left(1 - \frac{\eta}{t\langle\eta\rangle}\right)$$

در نتیجه تابع توزیع گره‌ای با جذابیت η در t بی‌نهایت بدست می‌آید، در نتیجه داریم:

$$P(k) = \lim_{t\to\infty} P(k,t)$$

پس از انجام این کار مورد عجیبی که رخ می‌دهد این است $t+1$ حدودا برابر t است در نتیجه بدست می‌آید:

$$P(k-1,t) = P(k,t)$$

که منطقی نیست و برای ما عجیب است. و اگر این فرض را نکنیم داریم:

$$\frac{dP(k,t)}{dt} = \frac{\eta}{t\langle\eta\rangle}\big(P(k-1,t) - P(k,t)\big) = \frac{\eta}{\langle\eta\rangle}\ln\left(\frac{t}{\eta}\right) = -\ln\left(\frac{P_{k-l} - P_k}{P_{k-1} - P_1}\right)$$

$$\Rightarrow P_{k-1} - e^{-\frac{\eta}{\langle\eta\rangle}\ln(t)}(P_{k-1} - P_1)$$

$$\Rightarrow P_k = P_{k-1}\left(1 - t^{\frac{-\eta}{\langle\eta\rangle}}\right) - t^{\frac{-n}{\langle\eta\rangle}} P_1$$

در نتیجه در t بی‌نهایت همان نتیجه عجیب قبل را بدست می‌آوریم.

که این نتیجه در واقعیت آنقدر عجیب نیست زیرا در زمان بی‌نهایت به دلیل کوچک شدن اعضا در برابر کل شبکه منطقی است که احتمال اضافه شدن راس‌ها یکی شود.

c)

جذابیت نصف گره‌ها، $\eta = 2$ باشد و جذابیت نیمی دیگر $\eta = 1$. می‌توانیم از تابع بدست آمده از قسمت قبل استفاده کنیم، داریم:

$$P_{\eta=2}(k, t+1) = P(k-1, t) \times \frac{2}{t\langle\eta\rangle} + P(k, t) \times \left(1 - \frac{2}{t\langle\eta\rangle}\right)$$

$$P_{\eta=1}(k, t+1) = P(k-1, t) \times \frac{1}{t\langle\eta\rangle} + P(k, t) \times \left(1 - \frac{1}{t\langle\eta\rangle}\right)$$

که $\langle\eta\rangle = \frac{1}{2} \times 2 + \frac{1}{2} \times 1 = 1.5$ است.

توزیع درجه شبکه پس از مدت زمان کافی برابر است با:

$$P(k) = \lim_{t\to\infty}\left[\frac{1}{2} P_{\eta=1}(k, t) + \frac{1}{2} P_{\eta=2}(k, t)\right]$$

حل این معادلات و گرفتن حد با نزدیک شدن t به بی‌نهایت، توزیع درجه شبکه را فراهم می‌دهد.

سوال 1) کد زنی:

کد این بخش به فایل ضمیمه شده است.

ابتدا مقدار دهی اولیه را انجام می‌دهیم:

```
n = 4;
FinalN = 1000;
SpecialVert = {7, 25, 47, 60, 150};
CoreGraph = Graph[CompleteGraph[n + 1], VertexLabels -> Automatic]
SeedRandom[63456];
Eta = RandomReal[{0, 1}, FinalN];
Eta[[SpecialVert]] = {0.5, 0.8, 0.9, 0.6, 0.9}
```

که در آن n تعداد رئوس اولیه در نمودار.

FinalN: تعداد کل مراحل یا تکرار برای تکامل نمودار.

SpecialVert: فهرست رئوس هایی که ویژگی های خاصی خواهند داشت.

CoreGraph: نمودار کامل اولیه با n + 1 راس.

SeedRandom[63456]: دانه تصادفی را برای تکرارپذیری تنظیم می‌کند.

Eta: لیستی از اعداد واقعی تصادفی بین 0 و 1. برخی از مقادیر در Eta با مقادیر خاصی برای رئوس در SpecialVert جایگزین می‌شوند.

حال تابع اضافه شدن گره و تشکیل گراف را می‌نویسیم:

```
NodeAdd[g_, n_] := (Module[{Newg, x},
   x = VertexCount[g] + 1;
   Newg = VertexAdd[g, x];
   Newg = EdgeAdd[Newg,
     UndirectedEdge[x, #] & /@
      RandomChoice[Eta[[1 ;; x - 1]]*VertexDegree[g] -> VertexList[g], n]];
   Newg
  ])
```

NodeAdd تابعی است که یک نمودار g و یک عدد صحیح n می‌گیرد.

یک راس جدید به نمودار اضافه می‌کند و آن را به n راس موجود متصل می‌کند، با احتمال اتصال به یک راس متناسب با درجه آن.

ادامه‌ی کار گتشکیل گراف:

```
SeedRandom[65354];
{t, FinalGraph} = Timing[Nest[(NodeAdd[#, n] &) , CoreGraph, FinalN - 5]]; t
```

این تکامل گراف را با اعمال تکراری تابع NodeAdd ایجاد می‌کند.

تکامل در FinalGraph ذخیره می‌شود و زمان صرف شده برای تکامل در t ذخیره می شود.

رسم نمودار تابع توزیع درجه:

در انتها نتایج آن به صورت log-log خطی آورده شده است.

```
Histogram[VertexDegree[FinalGraph], ScalingFunctions -> {"Log", "Log"}]
```

حال به سراغ تکامل درجات می‌رویم:

```
GraphEvolution = NestList[(NodeAdd[#, n] &) , CoreGraph, FinalN - 5];
```

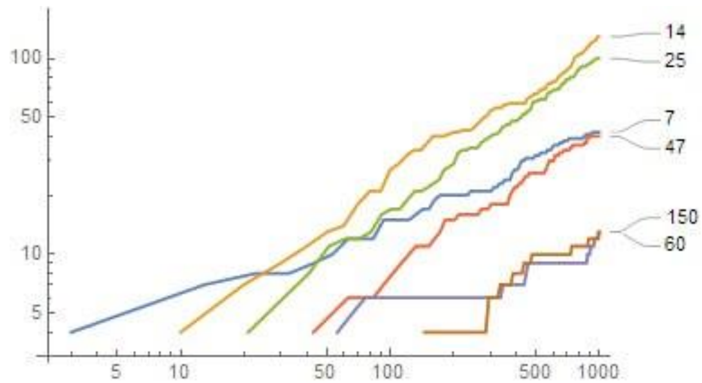
```
DegreeEvolution[v_, step_, n_] := (Module[{Steps},
   Steps = Range[v - (n + 1) + 1, FinalN - (n + 1) + 1, step];
   Transpose[{Steps, (VertexDegree[#, v] &) /@
      GraphEvolution[[v - (n + 1) + 1 ;; FinalN - (n + 1) + 1 ;; step]]}]
  ])
```

یک تابع DegreeEvolution را تعریف می‌کنیم که شاخص راس، اندازه گام و n را می‌گیرد. درجه راس مشخص شده را در مراحل مختلف در طول تکامل محاسبه می‌کند.

```
Ds = DegreeEvolution[#, 10, n] & /@ SpecialVert;
ListLogLogPlot[Ds, PlotLabels -> SpecialVert, Joined -> True]
```

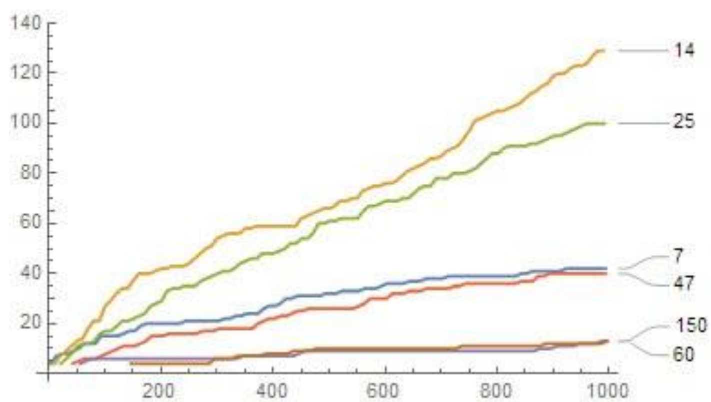
DegreeEvolution را روی رئوس در SpecialVert با اندازه گام 10 اعمال کرده. تکامل درجه را در مقیاس log-log ترسیم می‌کنیم.

نمودارها:



برای η های دلخواه، مشاهده می‌کنیم که لگاریتم درجات نسبت به لگاریتم زمان به صورت بالا است. همانطور که می‌بینیم، برای راس 14 که فیتنس بیشتر از راس 7 دارد، رشد سریع‌تری داشته و با آنکه راس 7 بسیار سریع‌تر وارد شبکه شده اما راس 14 از آن پیشی گرفته. این مورد را برای راس 25 نیز مشاهده می‌کنیم. از طرفی راس 150 که فیتنس خیلی بالایی دارد به دلیل بسیار دیر وارد سیستم شدن نتوانسته خود را به رئوسی با فیتنس کمتر برساند.

نمودار خطی نمودار بالا را اینجا مشاهده می‌کنیم:



با فیت کردن خط بر روی نمودار لوگ- لوگ اول، شیب خط‌ها را به ترتیب بدست می‌آوریم:

$$n = 7 \quad a_7 = 0.5$$

$$n = 14 \quad a_{14} = 0.7$$

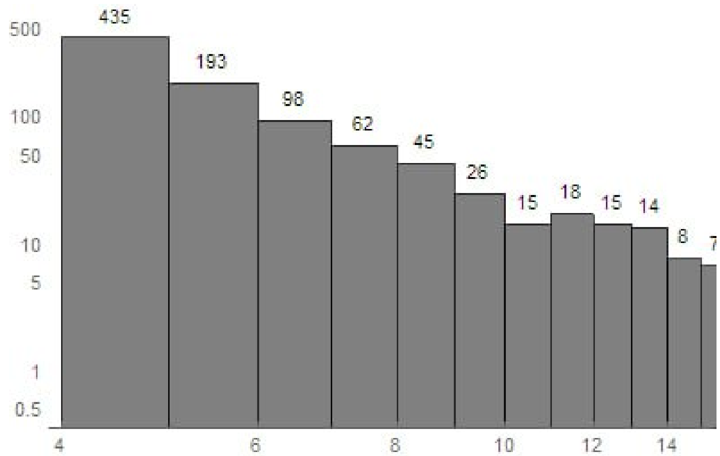
$$n = 25 \quad a_{25} = 0.8$$

$$n = 47 \quad a_{47} = 0.7$$

$$n = 60 \quad a_{60} = 0.3$$

$$n = 150 \quad a_{150} = 0.7$$

برای تابع توزیع درجه به صورت لوگ لوگ بدست آوردیم:



برای خطی این نمودار داریم: