Véletlen fizikai folyamatok

7. beadandó

Márton Tamás

PJF19C
martontamas@caesar.elte.hu

1. feladat

Feladat leírás.

A 8. heti előadáson megoldottuk az Erdős-Rényi gráf növekvő hálózatként értelmezett dinamikai általánosítását. Meghatároztuk a hálózat fokszámeloszlását, $P_k(t)$-t, amelyre Poisson eloszlást kaptunk. Ebből következett, hogy a fokszám átlaga egyenlő a szórásnégyzetének átlagával, $\langle k\rangle = \langle k^2\rangle - \langle k\rangle^2$. A feladat most ennek az eredménynek a deriválása a fokszámeloszlás meghatározása nélkül. Ehhez vissza kell térnünk a $P_k(t)$-ra vonatkozó master egyenlethez [8. előadás, (8-9) egyenlet], s származtatnunk kell a $\langle k\rangle$-ra és $\langle k^2\rangle$-re vonatkozó differenciálegyenleteket (a származtatás menetére lásd a 7. előadást) és persze meg kell találnunk a megoldásukat is.

Az adott esetben a modell elég egyszerű ahhoz, hogy differenciálegyenletek zártak legyenek, azaz ne tartalmazzák k magasabb momentumainak átlagát, s megoldásuk se legyen nehéz.

Feladat megoldása.

A megoldáshoz a $P_k(t)$ re felírt master egyenletből indulunk ki:

$$\frac{\partial}{\partial t}P_k = P_{k-1} - P_k$$
$$\frac{\partial}{\partial t}P_0 = -P_0$$

A két alábbi momentum érdekel minket:

$$\langle k\rangle = \sum_{k=0}^{\infty} kP_k(t)$$
$$\langle k^2\rangle = \sum_{k=0}^{\infty} k^2P_k(t)$$

Első lépésben deriváljuk le $\langle k\rangle$-t, ahol most $k = 1$-gyel kezdődik:

$$\langle\frac{\partial}{\partial t}k\rangle = \sum_{k=0}^{\infty} k\frac{\partial}{\partial t}P_k(t) = \sum_{k=1}^{\infty} kP_{k-1}(t) - \sum_{k=1}^{\infty} kP_k(t).$$

Átalakítom az egyenletet az alábbi egyenletek alapján:

$$\sum_{k=1}^{\infty} kP_{k-1}(t) = \langle k \rangle + 1$$
$$\sum_{k} P_k = 1.$$

Így az egyenletünk, ha mindent visszaírok, az alábbi egyenletet kapok:

$$\langle \frac{\partial}{\partial t} k \rangle = \langle k \rangle + 1 - \langle k \rangle = 1. \tag{1.1}$$

Majd a végeredményért elvégzem az integrálást:

$$\langle k \rangle = t. \tag{1.2}$$

Következő lépésben $\langle \frac{\partial}{\partial t} k^2 \rangle$ -t határozzuk meg:

$$\langle \frac{\partial}{\partial t} k^2 \rangle = \sum_{k=1}^{\infty} k^2 \frac{\partial}{\partial t} P_k(t) = \sum_{k=1}^{\infty} k^2 P_{k-1}(t) - \sum_{k=1}^{\infty} k^2 P_k(t), \tag{1.3}$$

ahol a második tagot felismerjük, hogy $\langle k^2 \rangle$, a többit pedig átalakítom:

$$\sum_{k=1}^{\infty} k^2 P_{k-1}(t) = \sum_{k=1}^{\infty} (k-1)^2 P_{k-1}(t) + \sum_{k=1}^{\infty} (2k-1)^2 P_{k-1}(t) =$$
$$= \sum_{k=1}^{\infty} k^2 P_{k-1}(t) + 2\sum_{k=1}^{\infty} (k-1)^2 P_{k-1}(t) + \sum_{k=1}^{\infty} P_{k-1}(t) =$$
$$= \langle k^2 \rangle + 2\langle k \rangle + 1.$$

Ha mindent visszaírok és átrendezek:

$$\langle \frac{\partial}{\partial t} k^2 \rangle = 2\langle k \rangle + 1. \tag{1.4}$$

Ebbe visszaírva $\langle k \rangle$-ra kapott értéket és kiintegrálom:

$$\langle k^2 \rangle = t^2 + t. \tag{1.5}$$

Ha a kapott eredményeket beírom a feltett kérdésben szereplő egyenletbe, akkor megkapjuk a keresett értéket:

$$t = \langle k \rangle = \langle k^2 \rangle - \langle k \rangle^2 = t^2 + t - t^2 = t. \tag{1.6}$$

2. feladat

Feladat leírás.

Az egyik legegyszerűbb hálózatnövekedési dinamika a *véletlen rekurzív fát* állítja elő. A hálózat növekedése abból áll, hogy minden lépésben egy új csúcsot kötünk egy éllel a meglévő csúcsok egyikéhez, egyenlő valószínűséggel bármelyikhez. Tehát a hálózat egy csúccsal indul, s első lépésben csak hozzákötjük a második csúcsot. A második lépésben már két csúcs közül választ a bejövő harmadik csúcs. A választás a két csúcs közül egyenlő (tehát 1/2-1/2) valószínűséggel történik. Hasonlóan, a negyedik csúcs 1/3-1/3-1/3 valószínűséggel kötődik a meglévő három csúcs egyikéhez. Ez a növekedési dinamika folytatódik (rekurzív módon, innen a rekurzív fa elnevezés), amíg felépül egy $N \gg 1$ csúcsból álló fa.

Szimuláljuk a fentiekben definiált rekurzív fát, s oldjuk meg az alábbi feladatokat.

(i) Próbáljunk elgondolkodni azon, hogy milyen jellegű fokszámeloszlásra számíthatunk, s miért fog ez különbözni az Erdős-Rényi dinamika eredményétől!

(ii) Határozzuk meg a csúcsok fokszámeloszlását ($P_k = N_k/N$, ahol N a csúcsok száma, N_k pedig a k éllel rendelkező csúcsok száma).

(iii) Vizsgájuk mekkora N kell ahhoz, hogy az eloszlásfüggvény P_k hibája kisebb legyen mint 10% minden $k \leq 5$-re. Figyelem, az egzakt eloszlásfüggvényt 9. heti jegyzetben kiszámoljuk, az eredmény $N \to \infty$-re a következő alakú $P_k^e = 1/2^k$ $(k \geq 1)$!

(iv) Határozzuk meg az átlagos fokszámot mind elméletileg, mind pedig a szimulációkból!

(v) Találjuk meg a maximális fokszámú csúcsot a fenti szimulációkban generált hálózatokban. Többször megismételve a szimulációkat $N = 100$, 1000 és 10000 esetére, határozzuk meg a maximális fokszám átlagát, $\langle k_{max} \rangle$-t! Látunk-e valamilyen funkcionális trendet az eredményekben?

Feladat megoldása.

(i)
Az Erdős-Rényi gráfban végig konstans a csúcsok száma, emiatt a mi esetünkben $\frac{1}{2^k}$ féle eloszlás alakul ki, míg az Erdős-Rényi gráfnál Gauss-eloszlás.
Valamint a mi esetünkben folyton növeljük a csúcsok számát.

A szimulációs feladatot most is *python* programnyelven oldottam meg. Első lépésben létrehoztam N változót a lépések számát, amit majd változtatok a különböző lépésszámokra. Valamint a csúcsoknak és a fokszámoknak hoztam létre a tömböket és feltöltöttem a csúcsok tömbjét 1-től N-ig egyesével egész számokkal.

```
%pylab inline
N = 10000

csucsok = numpy.zeros(N)
fokszamok = numpy.zeros(N)
for i in range (0,N):
        csucsok[i] = i+1
```

Az első és második csúcsot rögtön beállítom, hiszen azokat egyértelmű, hogy melyikhez tudjuk kötni őket.

```
fokszamok[0] = 1
fokszamok[1] = 1
```

Majd betettem az új csúcsokat a hálózatba 2-től, N-ig. Fontos, hogy mikor beteszem az új csúcsot, akkor egy random csúcsot választok ki és ahhoz kötöm.

```
for i in range (2,N):
        random_kivalasztott_csucs = randint(0,i)
        fokszamok[random_kivalasztott_csucs] += 1
        fokszamok[i] += 1
```

Majd kigyűjtöttem a fokszámokat és azok előfordulásának számát.

```
fokszameloszlas = []
fokszamokfajtaja = []
for i in range (0,N):
        if fokszamok[i] not in fokszamokfajtaja:
                fokszamokfajtaja.append(fokszamok[i])
                fokszameloszlas.append(1)
        if fokszamok[i] in fokszamokfajtaja:
                fokszameloszlas[fokszamokfajtaja.index(fokszamok[i])] += 1
```

Viszont sorrendbe kellett rendeznem a fokszámokat és a hozzá tartozó gyakoriságukat, mert az első kigyűjtés nem így hajtottam végre.

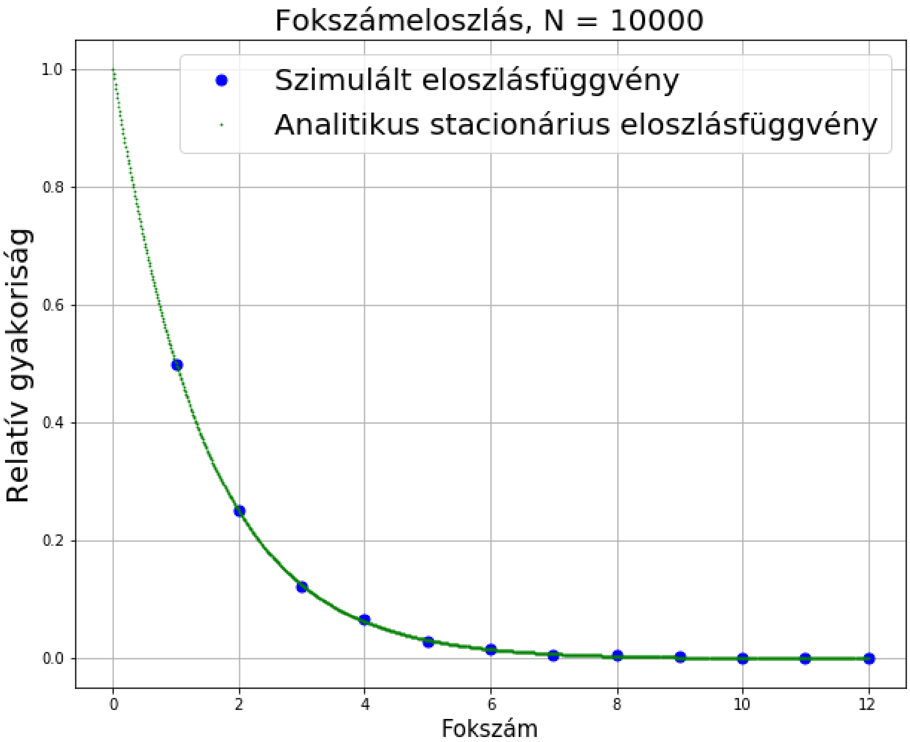
```
rendezett_fokszameloszlas = []
rendezett_fokszamokfajtaja = []
for i in range(1,len(fokszamokfajtaja)+1):
        for j in range(0,len(fokszamokfajtaja)):
                if fokszamokfajtaja[j] == i:
                        rendezett_fokszamokfajtaja.append(i)
                        rendezett_fokszameloszlas.append(fokszameloszlas[j])
```

(ii)
Majd elkészítettem az eloszlásfüggvényt, mely: $P_k = \frac{N_k}{N}$, ahol N_k a fokszámok gyakorisága, N pedig az összes csúcsszám.

```
rendezett_fokszameloszlas = numpy.array(rendezett_fokszameloszlas)
rendezett_fokszamokfajtaja = numpy.array(rendezett_fokszamokfajtaja)
eloszlasfuggveny = rendezett_fokszameloszlas / N
```

Az előadáson levezetett érték stacionárius esetben $P_k^{stac} = \frac{1}{2^k}$, amit számoltam a programmal is és ábrázoltam is n = 10000.

```
def analitikus_mo(k):
        return 1/2**k
```



2..1. ábra. Szimulált és analitikus fokszámeloszlás példa, N = 10000 esetében.

(iii.)
A hiba meghatározásához a következő lépéseket végeztem:

- a k értékhez ismerjük a relatív gyakoriságot a szimulációból.
- az analitikus megoldást is kiszámolom a k értékekhez.
- majd megnéztem, hogy hány százaléka a szimulált érték az analitikusnak.
- megállapítom, hogy hibahatáron belül van-e az érték.

A vizsgálat közben, mindig kiírattam, hogy melyik k érték esetén van nagyobb, mint 10%, és így tudtam, hogy mennyit kell változtatnom N értékét.

```
arany = numpy.zeros(10)
if(eloszlasfuggveny.shape[0] < 10):
        meddig = eloszlasfuggveny.shape[0]
if(eloszlasfuggveny.shape[0] >= 10):
        meddig = 10
for k in range(0,meddig):
        arany[k] = eloszlasfuggveny[k] / analitikus_mo(k+1) * 100
        if(arany[k] < 90 or arany[k] > 110):
                print("k=" + str(k+1) +
                " eseten az elteres : " +
                str(abs(arany[k]-100)) + "%")
```

N értékét 100-tól növeltem folyamatosan és úgy találtam, hogy mikor elértem a 200000 es N számot, akkor már teljesült a 10% alatti hibahatár*(200000 felett is)*.

(iv)
Az átlagos fokszámot a szimuláció alapján, úgy kaptam meg, hogy kiátlagoltam a fokszámokat.

```
atlagfokszam = 0
for i in range (0,shape(rendezett_fokszamokfajtaja)[0]):
        atlagfokszam += rendezett_fokszamokfajtaja[i] * rendezett_fokszamelosz
atlagfokszam = atlagfokszam / N
print("Atlagos fokszam szimulaciobol: " + str(atlagfokszam))
```

Hogy meg tudjam adni az átlagos fokszámot hibával együtt, lefuttattam a szimulációt különböző N-ekre, majd kiátlagoltam.
Az átlag hibája:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \langle x\rangle)^2 \rightarrowtail \sigma = \sqrt{\sigma^2}. \tag{2.1}$$

N	$\langle k\rangle$ átlagos fokszám
200000	2.000845
300000	2.000415
500000	2.000376
1000000	2.000169
10000000	2.0000225
Átlag hibával	2.0003655 $\pm$ 0.0002788

1. táblázat. Átlagos fokszámhoz különböző N esetekre mérések, majd az átlagos fokszám hibával.

A következő lépésben elméleti úton számolom ki $\langle k\rangle$ értékét.

$$\sum_{k=0}^{\infty}\frac{k}{2^k} = \sum_{k=0}^{\infty}k\left(\frac{1}{2}\right)^k \tag{2.2}$$

Valamint a végtelen mértani sorról tudjuk azt:

$$\sum_{k=0}^{\infty}q^k = \frac{1}{1-q}, \quad ha|q| < 1, \tag{2.3}$$

majd deriválom mindkét oldalt:

$$\sum_{k=0}^{\infty} k \cdot q^{k-1} = \frac{1}{(1-q)^2}, \tag{2.4}$$

majd átindexeljük:

$$\sum_{k=0}^{\infty} (k+1) \cdot q^{k} = \frac{1}{(1-q)^2}, \tag{2.5}$$

még egy kicsit alakítunk rajta:

$$\sum_{k=0}^{\infty} (k+1) \cdot q^{k} = \sum_{k=0}^{\infty} kq^k + \sum_{k=0}^{\infty} q^k, \tag{2.6}$$

Felhasználom az előző összefüggéseket(2, 2):

$$\sum_{k=0}^{\infty} kk^q = \frac{1}{(1-q)^2} - \frac{1}{1-q} = \frac{q}{(1-q)^2}. \tag{2.7}$$

Most nekünk $k = 1/2$, tehát $\langle k \rangle$ átlagos fokszáma:

$$\langle k \rangle = \frac{1/2}{(1-1/2)^2} = 2. \tag{2.8}$$

Tehát visszakaptam az előadáson kiszámolt megoldást, valamint szimulálva is ezt az értéket kaptam.

v.

Maximális fokszámot az alábbi kóddal kaptam meg:

```
index = where(fokszamok == max(fokszamok))
print("Az ennyiedik csucs(ok)nak van maximalis fokszama:"
+ str(csucsok[index[0]]) +
 " ami ekkora fokszamu: " + str(max(fokszamok)))
```

A számolás eredményeit az alábbi táblázatok tartalmazzák:

N	$\langle k_{max} \rangle$ maximális fokszám
100	8
100	8
100	9
100	8
100	7
100	6
100	6
100	6
100	9
100	9
Átlag hibával	7.6 ± 1.2

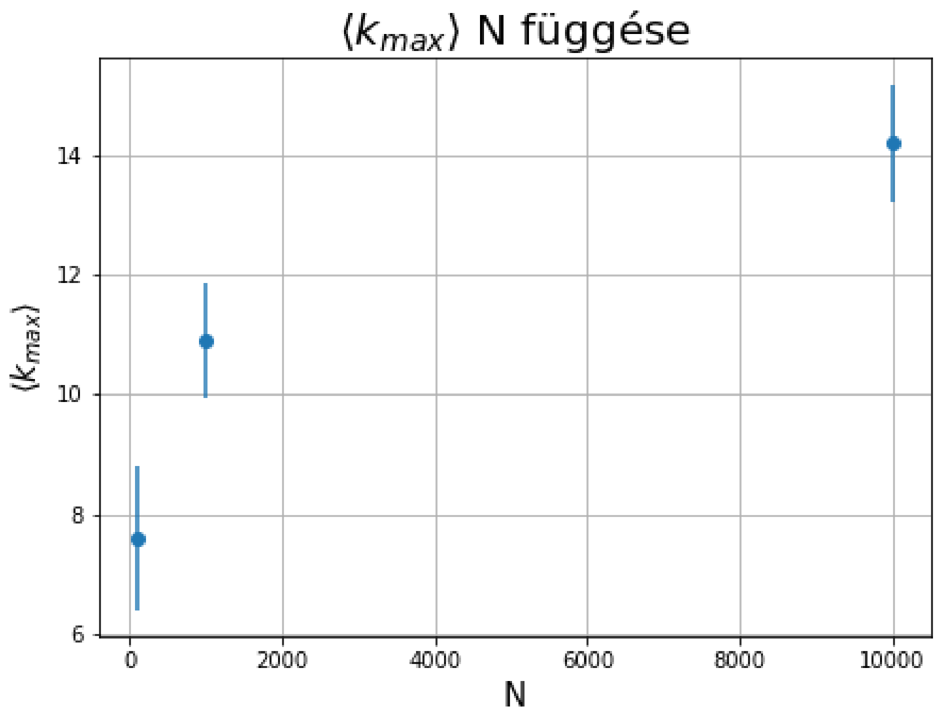
2. táblázat. $\langle k_{max} \rangle$ fokszámhoz különböző N = 100 esetekre.

N	$\langle k_{max} \rangle$ maximális fokszám
1000	10
1000	12
1000	11
1000	10
1000	11
1000	11
1000	11
1000	10
1000	10
1000	13
Átlag hibával	10.9 ± 0.943

3. táblázat. $\langle k_{max} \rangle$ fokszámhoz különböző N = 1000 esetekre.

N	$\langle k_{max} \rangle$ maximális fokszám
10000	13
10000	14
10000	14
10000	13
10000	15
10000	15
10000	15
10000	14
10000	16
10000	13
Átlag hibával	14.2 ± 0.979

4. táblázat. $\langle k_{max} \rangle$ fokszámhoz különböző N = 10000 esetekre.



2..2. ábra. $\langle k_{max} \rangle$ maximális fokszám N függése.