

ÖSSZETETT HÁLÓZATOK OKTATÁSA A CXNET PROGRAMMAL

TEACHING COMPLEX NETWORKS WITH THE CXNET PROGRAM

Horváth Árpád

Óbudai Egyetem, Alba Regia Egyetemi Központ, Székesfehérvár

ÖSSZEFOGLALÁS

A tudomány több területén fordulnak elő nagy gráfokként ábrázolható rendszerek, melyek tulajdonságai jelentősen eltérnek a véletlen gráfokétól. Ezeket összetett hálózatoknak nevezzük. Az összetett hálózatok fogalmi és vizsgálati módszerei jogosultságot nyertek többek között az egyes fehérjék és gének szerepének meghatározásában, az agy felépítésének, az internet sérülékenységének és a fertőzések terjedésének vizsgálatában. Hogy könnyen lehessen vizsgálni ilyen hálózatokat akár a közép- vagy felsőfokú oktatásban is, kifejlesztettünk egy cxnet nevű szoftvert, mely lehetővé teszi, hogy interaktív módon, vagy további program írásával létrehozzunk, vizsgáljunk hálózatokat. Az előadás során ismertetjük a szoftver használatát, különös tekintettel a program nemrég kifejlesztett részére, mely lehetőséget ad előre megadott tulajdonsággal rendelkező hálózatok generálására.

ABSTRACT

In many fields of the science there are systems that can be represented as large graphs. These systems are called complex networks. Complex networks can be used to determine the role of the proteins and genes, and investigate the structure of brain, the robustness of the internet and the infection spreading. We developed a software to create, investigate complex networks, read them from archive files, and plot their degree distribution interactively.

BEVEZETÉS

Az összetett hálózatok tudománya az 1990-es évek második felétől jelentős interdiszciplináris tudományterületté nőtte ki magát. Ezt több biológiai, információs, technikai és szociológiai hálózat teljes vagy részleges feltérképezése, a növekedésüket befolyásoló törvényszerűségek felismerése [1,2], és tulajdonságaik gyakorlati következményeinek felismerése tette lehetővé [3,4].

A cxnet programcsomag a Python programozási nyelvhez [9] tartalmaz két modult, az alapját képező `igraph` [10] és `pylab` modulok segítségével lehetőséget adnak a nagy hálózatok (gráfok) vizsgálatára. A szerző a cxnet használatával informatikushallgatók egy szabadon választható tárgyának keretében oktat az Óbudai Egyetemen egy olyan gépteremben, ahol a gépeken Debian GNU/Linux operációs rendszerek is futnak.

A CXNET TELEPÍTÉSÉRŐL ÉS HASZNÁLATÁRÓL

A csomagot kiterjedtebb módon Linux rendszereken (Ubuntu 11.04, Debian 6.0) teszteltük, de legtöbb lehetősége Windows rendszer alatt is működik. A cxnet teljes körű angol nyelvű dokumentációja (használat és telepítés) és némi magyar nyelvű segítség megtalálható a <http://arek.uni-obuda.hu/cxnet> oldalról kiindulva.

A cxnet legtöbb lehetősége az igrph nevű C-ben írt függvényeket hívogató, gyors hálózatzvizsgáló modulra épül. A grafikonok (például eloszlásfüggvény) kirajzolásához a cxnet a pylab modult használja, melyhez a matplotlib és a numpy csomagok telepítése szükséges. A csomagfüggőségi hálózat létrehozásához a Python apt modulja szükséges. Az interaktív használat során a legkellemesebb felületet tapasztalataink szerint az ipython parancsértelmező szolgáltatja.

A tárgy oktatásához tanmenetet dolgoztunk ki hozzá kapcsolódó programozási feladatokkal, és a tananyag számonkérésére interneten elérhető tesztek a Moodle rendszeren belül. Szerencsére több kiváló összefoglaló cikk szabadon elérhető angol nyelven [5-7]. A terület jó összefoglalását találhatjuk Mark Newman: Networks című könyvében [8].

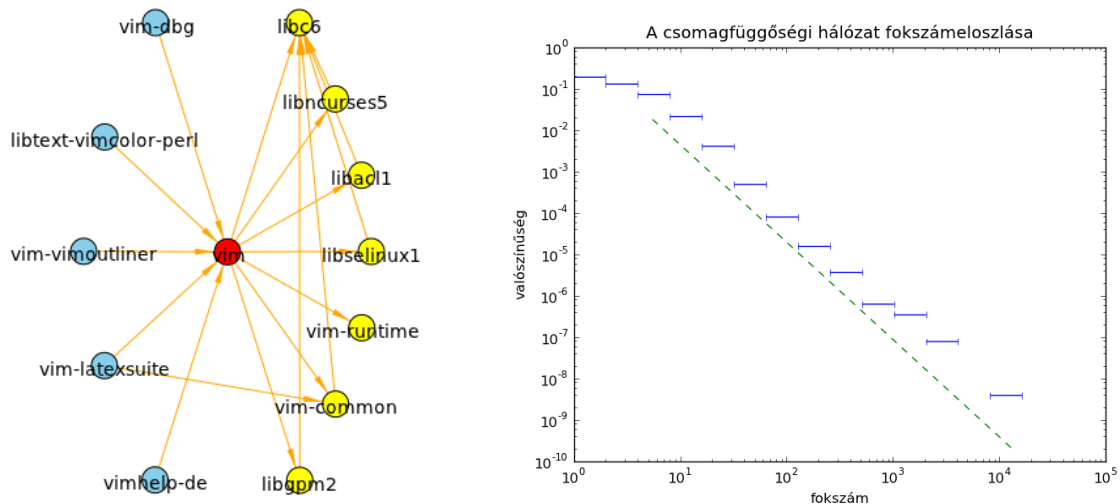
Az érdeklődés felkeltésére kiválóan alkalmas Barabási Albert-László Behálózva című ismeretterjesztő könyve [13], és a Mindentudás Egyetemének hálózatokkal kapcsolatos előadásai, melyek a <http://arek.uni-obuda.hu/cxnet> oldalon összegyűjtésre kerültek. Ezen az oldalon további, az oktatásban jól felhasználható információk és rövid magyar nyelvű videók találhatóak a Pythonról, a cxnetről, az igrph-ról, és a pylabról.

A CSOMAGFÜGGŐSÉGI HÁLÓZAT ÉS A FOKSZÁMELOSZLÁS

A cxnet csomag két Python programokban használható modult tartalmaz. Ebben és a következő szakaszban megmutatjuk a két modul néhány lehetőségét.

A cxnet modulban létrehozhatóak például a deb szoftvercsomag-fájlokat használó Linux terjesztésekben (Debian, Ubuntu) a csomagok függőségi hálózata, betölthetőek fájlokból egyéb hálózatok, és megvizsgálható azok fokszámeloszlása.

A Linux rendszerek többsége előre fordított szoftvercsomagokat használ a programok telepítéséhez. Ezek a szoftvercsomagok tipikusan internetes tárolókban találhatóak a róluk szóló információkkal együtt: jelen pillanatban milyen csomagok érhetőek el a tárolókban, és melyik csomag telepítéséhez melyik másik csomagokat szükséges telepíteni, azaz más szóval melyik másik csomagoktól függ. A cxnet csomag képes ezekből az információkból nagyjából 10 másodperc alatt egy hálózatot létrehozni, melyet korábban részletesen elemeztünk [12]. A cxnettel készült például az alábbi ábra a vim szövegszerkesztő csomagjának közvetlen környezetéről. Látható, hogy maga a vim közvetlenül függ a libc6 csomagtól, de a vim a libncurses5 csomagon keresztül közvetve is függ a libc6 csomagtól. Az 1. ábra jobboldalán ábrázoltuk a hálózat fokszámeloszlását, azaz, hogy a csúcsok hány százaléka rendelkezik egy adott fokszámmal. Az ábrán szaggatott vonallal egy $-2,35$ -ös kitevőjű hatványfüggvény látható, melynek egyenese jó közelítéssel párhuzamos az eloszlásfüggvénnyel 10-nél nagyobb értékekre, tehát a csomagfüggőségi hálózatot joggal tekinthetjük hatványfüggvény eloszlásúnak, más néven *skálafüggetlen hálózatnak*.



1. ábra. A bal oldali ábrán a vim csomag látható középen pirossal, a tőle függő csomagok balra (kék), és azok, amelyekről a vim csomag függ, jobb oldalon (sárga). A jobb oldali ábrán a csomagfüggőségi hálózat fokszámeloszlása látható logaritmusos bineléssel (kék szakaszok), valamint egy $-2,35$ -ös kitevőjű hatványfüggvény (szaggatott vonal).

AZ MFNG HÁLÓZATGENERÁLÓ MODUL

A cxnet csomag másik modulja egy úgynevezett multifraktálokra alapuló módszer [14], amellyel *egy előre megadott tulajdonságú hálózathoz egyre közelebb és közelebb kerülünk*. A hálózat csúcsai között a kapcsolódás valószínűségét egy kapcsolódási mérték adja meg. Ezt egy egyszerűbb mértékből, az úgynevezett generáló mértékből kapom meg. A generáló mértékben m -szer m darab p_{ij} -vel jelölt valószínűség található, melyet egy $m \times m$ -es mátrix elemeinek tekinthetünk. A mértéket a Descartes-féle koordináta-rendszer $[0,1) \times [0,1)$ egységnégyzetén definiáljuk. Annak mindkét tengelyét m részre osztjuk úgy, hogy az osztópontok mindkét tengelyen azonosak, ezáltal m -szer m darab kis téglalapot kapunk. Ezekhez a téglalapokhoz rendeljük a fenti valószínűségeket.

A generáló mértékből úgy kaphatom meg a kívánt n csúccsal rendelkező hálózatot, hogy először egy itt nem részletezett módszerrel egy hozzá hasonló, csak több téglalapból álló mértéket, az úgynevezett kapcsolódási mértéket hozok létre belőle. Majd minden csúcsához egy véletlen számot rendelek a $[0,1)$ intervallumból egyenletes eloszlással, majd minden egyes csúcspár esetén megnézem, hogy a két véletlen szám által meghatározott koordinátákban milyen valószínűség szerepel a kapcsolódási mértékben, és ezzel a valószínűséggel élt húzok a két csúcs között.

Hogyan kapom meg a kívánt hálózat generáló mértékét? A program először egy olyan generáló mértékkel kezd, amelyben mindkét tengely m egyenlő részre van osztva, és minden valószínűség megegyezik. Ezután a generáló mértéket elkezdem módosítani, és megnézem, hogy hogyan változtak a tulajdonságai, mennyivel került közelebb az elérni kívánthoz. Ehhez egy energiafüggvényt definiálunk úgy, hogy mennél jobban közelít a hálózat tulajdonsága a kívánt hálózathoz, annál kisebb legyen az energiája a generáló mértéknek. A módosítás lépéseiben a program felváltva változtatja a generáló mérték valószínűségeit, illetve az oldalak osztópontjait. A módosítás után az új generáló mértéket elutasítjuk, vagy elfogadjuk,

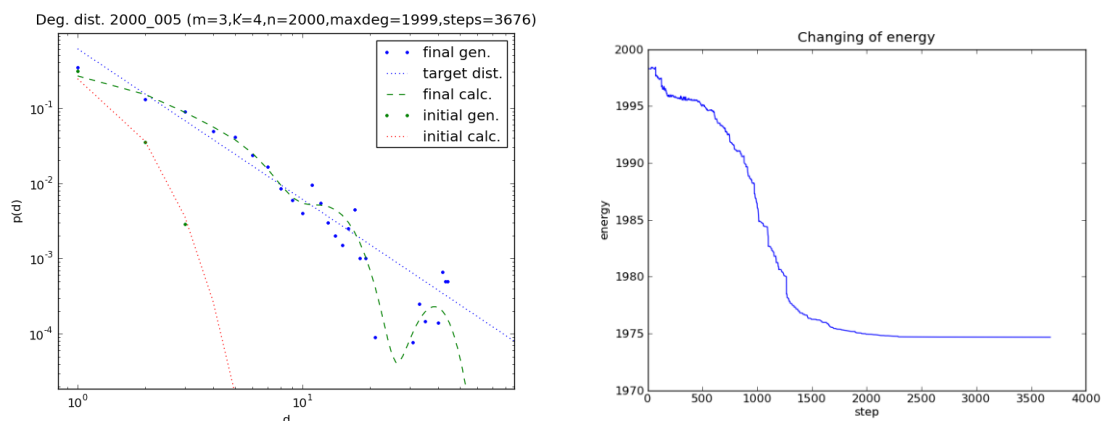
az első esetben az előzőleg elfogadott mértékkel, utóbbi esetben az új generáló mértékkel ismételve a lépéseket.

Mikor fogadjuk el a kapott generáló mértéket? Erre a Metropolis-algoritmust alkalmazzuk. Ha a kapott generáló mérték energiája kisebb lesz az előzőnél, akkor mindenképpen elfogadjuk. Egyéb esetben még egy paramétert figyelembe veszünk a döntésnél, egy hőmérsékletnek nevezett T értéket. Ezt az értéket a generálás során folyamatosan csökkentjük, két lépésenként adott, 1-nél kisebb számmal szorozva. A lépéseket egy adott hőmérsékletet eléréséig folytatjuk. Ha az új generáló mérték (E') energiája nagyobb, mint a pillanatnyilag elfogadotté (E), akkor

$$P(T) = \exp\left(-\frac{E'-E}{T}\right) \quad (1)$$

valószínűséggel elfogadom, $1-P(T)$ valószínűséggel elutasítom. A generálás során tehát a hőmérsékletet csökkentve változtatom a generáló mérték tulajdonságait, és a fenti algoritmus szerint elfogadom vagy elutasítom a kapott mértéket. Ezt sokszor megismétlem egyre csökkenő hőmérsékletértékkel, így a kapott generáló mértékhez tartozó hálózat közelít a kívánt tulajdonságú hálózathoz. A generáló mértékből bármikor létrehozhatok egy hálózatot a kapcsolódási mértéken keresztül, de a hálózat létrehozására nincsen szükség a generálás lépései során, az energia és a hálózat várható tulajdonságai kiszámíthatóak, ha ismerjük a generáló mértéket, az iterációk K számát és a hálózat csúcsainak n számát.

Az 2. ábra bal grafikonja egy generálás kezdeti és végső fokszámeloszlását ábrázolja. A megcélzott tulajdonság egy hatványfüggvény fokszámeloszlás volt, mely a kétszer logaritmikusan skálán a szaggatotttal jelölt egyenes. Mindkét ábrán jól látható, hogy a kezdeti fokszámeloszláshoz képest a generálás végén jóval közelebb került a hálózat a kívánt fokszámeloszlásúhoz.



2. ábra. A baloldali ábrán a fokszámeloszlás látható egy generálás kezdetén és végén. A céleloszlást kék pontozott vonal, a kezdeti generáló mértékből származó fokszámeloszlást piros pontozott vonal jelzi, a kapott eloszlást a szaggatott vonal. A különálló nagyobb pöttyök a kezdeti és végső generáló mértékből létrehozott egy-egy konkrét hálózat fokszámeloszlását jelölik. A jobboldali ábrán az energia változása látható a generálás során, a vízszintes tengelyen a lépés sorszáma, a függőlegesen az aktuális generáló mérték energiája látható.

ÖSSZEFOGLALÁS

Úgy gondoljuk, hogy a hálózatok vizsgálatának módszereit érdemes oktatni a felsőoktatás informatikával, biológiával és fizikával kapcsolatos területein. A cxnet és az igraph alkalmasak a hálózatok jellemzőinek bemutatására. A hálózatok vizsgálatának egyes területei akár középiskolai szakkörökben is elvégezhetőek.

IRODALOMJEGYZÉK

1. R. Albert, H. Jeong, AL. Barabási: Diameter of World-Wide-Web, Nature, vol. 401.
2. AL. Barabási, R. Albert: Emergence of scaling in random networks, Science, vol. 286
3. Z. Dezső, AL. Barabási: Halting viruses in scale-free networks, Physical Review E, vol. 65.
4. R. Pastor-Satorras, A. Vespignani: Epidemic spreading in scale-free networks, Physical Review Letters, vol. 86. pp. 3200-3203.
5. S. N. Dorogovtsev, A. V. Goltsev, J. F. F. Mendes: Critical phenomena in complex networks, REVIEWS OF MODERN PHYSICS, vol. 80, pp. 1275-1335.
6. R. Albert, AL. Barabási: Statistical mechanics of complex networks, REVIEWS OF MODERN PHYSICS, vol. 74, pp. 47-97.
7. MEJ. Newman: The structure and function of complex networks, SCIAM REVIEW, vol. 45, pp. 167-256.
8. M. Newman: Networks, Oxford University Press, 2010.
9. <http://www.python.org>
10. G Csárdi, T. Nepusz: The igraph software package for complex network research, InterJournal Complex Systems, 2006, pp. 1695
11. Á. Horváth: Studying complex networks with cxnet, Acta Physica Debrecina, vol XLIV.
12. Á. Horváth, Z. Trócsányi: Complex networks in the curriculum of computer engineers, International Symposium on Applied Machine Intelligence and Informatics, Herlany, 28-30 Jan. 2010.
13. Barabási A. L.: Behálózva: A hálózatok új tudománya, Helikon Kiadó, Budapest, 2008.
14. G. Palla, T. Vicsek, L. Lovász: Multifractal network generator, Proceedings of the National Academy of Sciences, vol. 107, no. 17, 2010.

SZERZŐ

Horváth Árpád, egyetemi adjunktus, horvath.arpad@arek.uni-obuda.hu, Székesfehérvár, Budai út 45.