KÖZÖSSÉGEK ÉS SZEREPÜK A KISVILÁG GRÁFOKBAN

BARTALOS ISTVÁN ÉS PLUHÁR ANDRÁS

KIVONAT. Összefoglaló írásunkban kísérletet teszünk a gráfokra kifejlesztett közösségkereső algoritmusok áttekintésére, egységesítésére és kiértékelésére. Bemutatjuk az eredményként előálló közösségi információ felhasználását a gráfos adatbányászatban és a gráfok segítségével végrehajtható modellezésben, melyeknek sikeres gyakorlati alkalmazásai vannak.

1. BEVEZETÉS

A *kisvilág gráfok* felfedezése jelentősen megváltoztatta, kibővítette a gráfelméleti kutatások irányát, lásd Barabási és Albert [2, 3]. Nemcsak ezek a gráfok különböznek a korábban vizsgált gráfoktól, hanem a velük kapcsolatban megfogalmazott kérdések és problémák is.

Nem könnyű feladat egy kisvilág gráf felépítéséhez szükséges információk összegyűjtése vagy éppen annak eldöntése, *hogyan* készítsünk a rendelkezésre álló adatokból gráfot, lásd Csernenszky és társai illetve Hidalgo és társai [13, 23]. Ugyanígy, bár számos próbálkozás történt, nincs minden igénynek eleget tevő modell véletlen kisvilág gráfok generálására sem, lásd Cami és Deo [11].

A valós alkalmazásokban fellépő méretek miatt időigényes algoritmusok nemigen használhatók, így jobbára meg kell elégedni egyszerűbb heurisztikákkal, melyek sokszor a fizikából kölcsönzött intuícióból erednek, lásd Barabási, Bollobás, Newman cikkei [3, 5, 28]. A szokásos jelölést követve egy G gráf ponthalmazát $V(G)$-vel, élhalmazát pedig $E(G)$-vel jelöljük. Ha az utóbbi *rendezett* párokat tartalmaz, akkor G *irányított*, és az élek *súlyozottak* is lehetnek.

A legtöbb további vizsgálat egyik alapvető feltétele a gráf pontjainak klasszifikációja, csoportokba rendezése. Ez történhet osztályozással, azaz $V(G)$-t felbontjuk $\{\mathcal{C}_i\}_{i=1}^m$ halmazok, ún. *klaszterek* diszjunkt uniójára. A másik megközelítésben nem kívánjuk meg sem a csoportjaink diszjunktságát, sem azt, hogy együtt kiadják $V(G)$-t. Ezeket az

1991 *Mathematics Subject Classification.* 05C82, 91D30, 68U20.

Key words and phrases. Domingos-Richardson model, infection model, data mining.

entitásokat szokás közösségeknek hívni; mi itt *közösség* alatt mindig ezeket értjük, míg az osztályozás elemeit klaszterekeknek hívjuk. Rengeteg erőfeszítés történt a klaszterek előállítására, vizsgálatára, illetve alkalmazására, részletesen lásd pl. Newman [28]. Annyit megjegyeznénk, hogy a klaszterek előállítására mind ún. *top down* (felülről lefelé) és *bottom up* (alulról felfele) építkező algoritmusokat javasoltak. Ezzel szemben a közösségek keresésére szolgáló algoritmusok jobbára az alulról építkezést használják, azaz kisebb közösségek növelésével próbálnak megfelelő eredményhez jutni.

A klaszterezés (és így a közösségkeresés is) elméletileg megalapozhatatlan Kleinberg [25] eredménye szerint, ezért a sokszor követett pragmatikus megoldás marad: veszünk egy ésszerűnek tűnő algoritmust, az eredményét definiáljuk klaszterekeknek/közösségeknek és megnézzük használhatóságát.

2. Néhány algoritmus

Három tipikus közösségkereső algoritmust tekintünk, melyek hasonló elven alapulnak. Az egyik első, ténylegesen használt algoritmus az N^{++}, Csizmadia és társai ill. Pluhár [8, 15, 31, 32]. A k-klikk perkolációs algoritmus, a CPM, az első széles körben ismert módszer, melyet Palla és társai [29] szintén valós feladatokra alkalmaztak. Az élek klaszterezése a harmadik - főként elméleti érdekességű Pluhár, Evans és Lambiote [31, 17].

2.1. Az $\mathbf{N}^{++}$ algoritmus. [32, 15] Ez egy generikus algoritmus egy tetszőleges

$$f : 2^{V(G)} \times V(G) \to \mathbb{R}$$

és $c : \mathbb{N} \to \mathbb{R}$ függvénnyel, ahol $f(A, x)$ jelenti az A közösség és az x csúcs kapcsolatának erősségét. Csatoljuk x-et A-hoz, ha $f(A, x) \geq c(|A|)$.

A **Build** szubrutin lentről felfelé építkezve megadja a közösségek $\mathcal{K}$ halmazának első közelítését.

A Build pszeudó kódja

begin(Build)
Input G, k, c (`max` k`-elemű` c`-közösségeket keresünk`)
Let $\mathcal{K} := V(G)$ (`kezdetben a csúcsok a közösségek`)
For $i = 1$ to k
$\forall A \in \mathcal{K}$, $x \in V(G)$ ha $f(A, x) \geq c(|A|)$ akkor tegyük $A \cup \{x\}$-t $\mathcal{K}$-ba.
Töröljük az összes olyan $A \in \mathcal{K}$-t, amelyre $A \subset B \in \mathcal{K}$ és $A \neq B$.
Print $\mathcal{K}$, „G legfeljebb k-elemű c-közösségei.”
end(Build)

A Build végrehajtása után a **Merge**-t használjuk a majdnem azonos közösségek összeolvasztására. Legyen C olyan gráf, amelyben $V(C) = \mathcal{K}$ és $(A, B) \in E(C)$, ha $A \cap B$ „elég nagy". Cseréljük ilyenkor $\mathcal{K}$-t $(\mathcal{K} \setminus \{A, B\}) \cup \{A \cup B\}$-ra. Ezután a C elemei legyenek a közösségek. A tapasztalat az alábbi értékeket javasolja. Jelentse a nagy a 60%-át a kisebb halmaz elemszámának. Az $f(A, x)$ értéke az x és A közötti egy és kettő hosszúságú utak számától függ. Tehát hogy megkapjuk az x-et tartalmazó közösségeket, elegendő keresni a $N^{++}(x) := N(N(x))$ halmazban, azaz legfeljebb a másod szomszédok között.

Néhány hasonló módszert sorol Fortunato [18].

2.2. k-klikkek perkolációja. Röviden CPM módszer, [29]. Itt $k \in \mathbb{N}$ adott, mint az algoritmus paramétere. Miután megtaláltuk az összes k-klikket G-ben, tekintjük azt a Q_k gráfot, melynek csúcsai ezen klikkek és $(A, B) \in E(Q_k)$ pontosan akkor, ha $|A \cap B| = k - 1$. A közösségek Q_k összefüggő komponensei klikkjeinek egyesítései lesznek.

2.3. Élek klaszterezése. [31, 17] Klaszterezzük valamilyen módon az élek halmazát. Az egyes klaszterek éleinek végpontjai lesznek a közösségek.

Ezek a módszerek különböznek a talált közösségek típusaiban és a számítási költségeikben is. Jóllehet az élek klaszterezését könnyű végrehajtani, használata mégis jelentős hátrányokkal jár. (pl. a kapott közösségek átfedése legfeljebb egy csúcspont mélységű.)

Az N^{++} és a CPM a legígéretesebbek algoritmusok; persze az implementációk minősége lényeges szempont. Kisvilág gráfokon mindkettő majdnem lineáris időben fut, ami természetes követelmény, ha valódi feladatokkal foglalkozunk.[1]

2.4. Egységes szemlélet. Vegyük észre, hogy a három felsorolt algoritmus család végrehajtása két lépésből áll. Először egy $\mathcal{F} = (V, \mathcal{H})$ hipergráfot határoznak meg, ahol $V = V(G)$ és $\mathcal{H} \subset 2^V$. A $\mathcal{H}$ elemei lesznek a közösségek *építőkövei*. A második lépésben $\mathcal{H}$-t alkalmas d távolságfüggvénnyel ellátva $\mathcal{M} = (\mathcal{H}, d)$ metrikus teret készítünk. Ezután valamilyen klaszterező algoritmussal $\mathcal{M}$ klasztereinek egy $\mathcal{C}$ halmazát kapjuk. Végül a keletkezett klasztereket V részhalmazaival azonosítjuk úgy, hogy egy $C_i \in \mathcal{C}$-re $K_i := \cup_{H \in C_i} H$, ahol K_i közösség megfelel C_i klaszternek.

[1]Ez csúcsok millióit jelenti. Az N^{++} elérhető a Sixtep szoftverrel, míg a klikkperkolációt a CFinder-rel próbáltuk ki. Ezennel megköszönjük a programok készítőinek, hogy tudományos célokra elérhetővé tették a szoftverüket.

A fenti algoritmusoknál $\mathcal{H}$ elemei (az építőkövek) rendre kis sűrűségű részgráfok, k-klikkek illetve élhalmazok. A köztük levő kapcsolatot leíró $\mathcal{D}$ gráfban pontosan akkor van él, ha a kapcsolat szoros. Az első esetben $(K_i, K_j) \in \mathcal{D}$, ha $|K_i \cap K_j|$ elég nagy, a másodikban, ha $|K_i \cap K_j| = k-1$, míg a harmadik esetben ez paraméter.

2.5. Központiság alapú közösségkeresések. Az előző alfejezet paradigmájába bele nem illő megoldások is lehetségesek. Costa [12] a nagy rangú pontok közül választ egy független halmazt; ezek lesznek a közösségek közepei, majd ρ sugarú gömböket képez körülöttük. Távolság függvénynek a G természetes metrikáját használja, amely a ρ paraméter értékétől függően átfedésekhez vezet(het). Egy másik megközelítésben Kovács és társai [26] először egy kifinomult hatás függvényt számolnak ki, amely a pontok központiságának mértéke. Ennek alapján nívófelületet képeznek, és a felület kiemelkedéseit azonosítják mint közösségeket.

3. Kiértékelés

Mivel a közösségek (vagy klaszterek) definíciói többé-kevésbé tetszőlegesek, Kleinberg [25], hasznosságuk mérésére is sokféle elgondolás született. Jóllehet ez alapvető kérdés, a kutatók nézőpontjai természetesen eltérőek. Az alábbiakban vázoljuk, hogyan lehet egy-egy közösség fogalom használhatóságát megállapítani. Egy *direkt módszer* közvetlenül hasonlítja össze az adódó közösségeket és a gráfról meglevő egyéb információnkat, míg az *indirekt módszerek* egy modell változójaként kezelik a közösségi információt, és az előrejelzés pontosításának a mértékén mérik ennek a hasznosságát.

3.1. Tapasztalatok és paraméterezés. Először futtatni kell az algoritmusokat, meg kell kapni az eredményeket és esetleg matematikai következtetéseket levonni bizonyos gráf osztályokról. Nagyon fontos az algoritmusok sebessége. Valódi sebességüket nem könnyű összehasonlítani, mivel ez erősen függ az implementációjuktól és a tesztgráfoktól (gyakorlati gráf avagy elméleti konstrukció).

Mindhárom algoritmus gyors és általában is a 2.4 alfejezetben leírt család algoritmusai hatalmas méretű problémák megoldására képesek. A 3.4 pontban még visszatérünk erre a kérdésre és néhány eredményt közlünk a futási időkről és a megoldások jóságáról, részletesen lásd Griechisch és Pluhár [22].

A klikk-perkolációs módszer figyelemre méltó mind elméleti, mind gyakorlati szemszögből nézve. Az Erdős-Rényi random gráfok kapcsán alaposan megvizsgálták Bollobás és Riordan [6] és a gyakorlatban is

használhatónak bizonyult, Adamcsek és társai [1]. Mindazonáltal a CPM néha túl nagy közösségeket ad és a paraméterezése is rejtélyes, hiszen hogyan döntjük el, milyen értéke legyen k-nak?

Az N^{++} algoritmus meglehetősen heurisztikus, elméleti vizsgálata nem kivitelezhető. Fő előnye a sebesség, a közösségek kis átmérője és a megbízhatóság.

Az él-klaszterező módszereket még kevéssé vizsgálták. Nyilvánvaló hátrányuk, hogy az általuk kapott közösségeknek legfeljebb egy közös elemük lehet. Valódi gráfoknál ez túl szoros feltétel.

Néhany benchmark gráfon kipróbáltunk a CPM és az N^{++} algoritmusokat, a tapasztalatokat Zachary híres gráfján illusztráljuk, lásd Zachary [35]. Ez a gráf a baráti kapcsolatokat írja le egy karate klubban, amely éppen a vizsgált időszakban vált ketté. Az egyik rész (A) a japán mesterrel maradt, míg a másik (B) az amerikai helyettesével tartott. A CPM $k = 3$ esetén három közösséget ad, rendre 3, 6 és 24 mérettel, míg $k = 4$-re szintén három közösség keletkezik, melyek mérete 4, 4 és 7. $k = 5$ esetén egyetlen 6 pontú közösség lesz. Itt a $k = 3$ és $k = 4$ esetek közösségeinek kombinálása tűnik jó megoldásnak, és a közösségek ekkor az A és B halmazok *belsejében* húzódnak. Az N^{++} algoritmus 12 közösséget ad, rendre a darabszámok/méretek: 4/3, 5/4, 1/6 és 2/7. Egyet kivéve a közösségek A vagy B belsejében vannak. A szakadás egy lehetséges magyarázata így éppen az A-t és B-t összekötő közösség felbomlása lehet.

3.2. Grafikus. A korai publikációk általában a gráf valamilyen vizuális formája alapján határozzák meg a közösségeket. A szem által végzett klaszterezések jónak bizonyultak. Az átlapolódó közösségek meghatározása már nehezebb, mert a vizualizáció már nem annyira kézenfekvő.

Egy lehetőség a különböző klaszterezések, közösségek összehasonlítására a gráf lerajzolása és a tetszés szerinti értékelése. A tapasztalat szerint a jó klaszterezések a szem szárára is kellemesek, az egy klaszterbe kerülő pontok többnyire közel vannak egymáshoz. A közösségek vizsgálatára már nem olyan egyszerű ilyen módon. Néhány ötlet segíthet, pl. a közösségek metszetgráfjának a megjelenítése. Az $I(G)$ metszetgráfban G közösségei a pontok és két pont akkor összekötött, ha a közösségek metszete nem üres, azaz $I(G) = (V(H), E(H))$, ahol $V(H) = \mathcal{K}$ és $(C_i, C_j) \in E(H)$ if $|C_i \cap C_j| > 0$. Hátránya ennek a megközelítésnek, hogy csak kis gráfokon használható és a klaszterek meghatározása mindig szubjektív.[2]

[2]Gráfok vizualizálására a *force directed* algoritmus bizonyult a legjobbnak. Azonban ez $O(n^2)$ időt igényel, ami megakadályozza használatát, ha n milliós nagyságú.

Ismét a Zachary gráfot tekintve, lásd Griechisch és Pluhár [22], a CPM egy nem összefüggő H gráfot ad. Az N^{++} által adott H metszetgráf informatívabb. Két sűrű részgráfból áll, melyeknek egy közös x pontja van, amely vágópont H-ban. Az x-nek megfelel egy négy pontból álló C_9 közösség, amely a japán mestert (1), a helyettesét (33) és a 3 illetve 9 számokkal címkézett embereket tartalmazza. (Ez a közösség különben az egyetlen, amelynek nem üres a metszete A-val és B-vel is.) $C_9 \cong K_4 \setminus e$, az egyetlen hiányzó él éppen a $(1, 33)$, ami érthető. Amikor a klub szakadása megtörtént, az elszakította 3 és 9 pontot, és ezzel megszűnt a C_9 közösség, amely addig kapocs lehetett a klubban. Kis fantáziával feltételezhető, hogy a eleve a 3-as és 9-es barátsága volt a klub kohéziójának az alapja, és mikor ez már nem viselte el a feszülséget és megszakadt, akkor az a klub végét is jelentette egyben.

3.3. Véletlen kisvilág gráfok. Sokféle módon lehet véletlen gráfokat generálni, melyek megragadják a kisvilág gráfok egy-egy lényeges tulajdonságát, lásd Barabási és Albert, Cami és Deo [2, 11]. Ezek közül a *Preferential Attachment* (PA) és a *Vertex Copy* (VC) modellekről szólunk részletesebben. Megjegyezzük, hogy másfajta megközelítések is vannak, pl. a véletlen metszetgráf modellt vizsgálja Stark [34].[3]

Mindkét modell rekurzívan definiált; egy már meglévő részgráfhoz vesz hozzá egy új x pontot, de az x szomszédságát másképp generálják. A PA modellben az x pont k új élet hoz, ezeket egymástól függetlenül és véletlenül kötjük a régi pontokhoz, egy y-hoz a $d(y)$ fokszámmal arányos valószínűséggel. A VC modellben egy régi s pontot választunk egyenletes eloszlással, és az új x ponttal az $N(s)$ pontjait p valószínűséggel, egymástól függetlenül összekötjük.

A tapasztalatok vegyesek és többet mondanak a modellekről, mint a CPM vagy N^{++} algoritmusokról. Az alábbiakban illusztráljuk a futási eredményeket két, nagyjából egy kategóriába tartozó gráfhalmazon, részletesen [22]. A gráfok 100 pontúak, a G_1 és H_1 gráfokat a PA modell adja, $|E(G_1)| = 192$, $|E(H_1)| = 358$, míg a G_2 and H_2 gráfok, amelyekre $|E(G_2)| = 151$ és $|E(H_2)| = 378$, a VC modell szerint állítottuk elő. A #C és #CO a klaszterek illetve közösségek számát jelenti, míg a k fejlécű oszlop a k méretű közösségek száma. A CPM esetében a k fejlécű oszlop viszont a az algoritmus k paraméterére utal, amely szerint a futás történt. A klasztereket Newman modularitás maximalizáló heurisztikája állította elő, lásd a következő alfejezetben.

[3] A metszetgráfokra a CPM hajlamos túl nagy közösségeket adni. A lehetséges javítás erre maximálni a közösségek átmérőjét az N^{++}-hoz hasonlóan.

Graph and Method	#C	#CO	3	4	5	6	7	> 7
G_1 / CPM	10	7	7					
G_1 / N^{++}	10	9	5	0	0	2	1	1
G_2 / CPM	9	17	13	4				
G_2 / N^{++}	9	22	8	7	2	4	1	0
H_1 / CPM	6	10	7	3				
H_1 / N^{++}	6	37	5	2	3	9	7	12
H_2 / CPM	6	24	4	8	6	6		
H_2 / N^{++}	6	26	8	3	2	5	1	7

3.4. Modularitás. A Newman modularitás [28] a G gráf és komponenseinek alábbi függvénye:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

ahol $m = |E(G)|$, A_{ij} a G adjacencia mátrixa, k_i az i-edik csúcs fokszáma, c_i a komponense és $\delta(c_i, c_j)$ a Kronecker szimbólum. A klaszterező algoritmusok alapulhatnak valamilyen matematikai vagy fizikai heurisztikán, mint pl. edge-betweenness (EB), eigenvectors (EV), label propagation (LP), spin glass (SG), walk trap (WT), vagy megpróbálják maximalizálni a modularitási függvényt az összes komponensek halmazán valamilyen mohó algoritmussal.

A modularitásra adott formula általánosítható közösségekre, Népusz és társai [27], ha s_{ij}-t írunk $\delta(c_i, c_j)$ helyett, ahol s_{ij} valamilyen i és j közötti hasonlósági mérték. (Jelen esetben u_i az i-edik pont valószínűségi eloszlása a közösségek fölött és $s_{ij} = \langle u_i, u_j \rangle$, de lehetne bármely $\|u_i - u_j\|$ norma is.)

Másrészt a közösségek *közvetlenül* is megkaphatók a modularitási függvény értékének maximalizálásával is, lásd [22]. Mivel egy kvadratikus célfüggvény maximalizálását kell elvégezni, ez a megközelítés csak kis gráfok esetén lehetséges, bár így is hasznos benchmark-okat ad. Egy másik út a optimum heurisztikákkal való megközelítése, csakúgy, mint a klaszterezés esetén. Egy másik tanulság, hogy a klaszterek és a közösségek szerkezete nem mérhető ugyanazzal a mértékkel, ezért további súlyozást kell használni. Az algoritmusok tesztelésének eredményeit a már jól ismert Zachary gráfon mutatjuk be. A klaszterezést klikk-perkoláció (CPM) követi, a klikkek mérete $k = 3$ and $k = 4$, and N^{++}. A futási idők másodpercben adottak, $\#C$ mutatja a klaszterek vagy közösségek számát (amelyiknek adott esetben értelmezett).

Method	Modularity	Running time	#C
EB	0.4013	0.0100	5
EV	0.3727	0.0000	3
Gr	0.3807	0.0000	3
LP	0.4020	0.0000	3
SP	0.4063	1.1500	6
WT	0.4198	0.0000	4
CPM 3	0.2438	0.012	3
CPM 4	0.2557		3
N^{++}	0.1947	0.6690	12

Algoritmusaink használhatóságát olyan hálózatokon ellenőrizhetjük, amelyek közösségei ismertek. Megfigyelhetők a különféle közösségi hálózatok (telekommunikációs, ismeretségi, Erasmus kapcsolatok gráfja stb.) működése közötti hasonlóságok és majdnem minden algoritmus hasznos észrevételeket eredményez. Megállapítható, hogy a közösségeket használó algoritmusok sokkal jobbak, mint a csak klasztereket használók.

3.5. Finomítások, idő és rendezések. Végezhetünk a grafikus módszerhez hasonló tanulmányokat is, ha van valamilyen, az éleken vagy a csúcsokon értelmezett függvényünk. Látunk néhány nagyon szubjektív, de mégis említésre méltó jelenséget.

i. Mindenekelőtt a klaszterek rendszerint jóval nagyobbak, mint a közösségek és a számuk is kevesebb.

ii. A közösségek száma akár a hatványtörvényt is követheti/követi, bár ezt ellenőrizni nem lehetséges.

iii. A közösségek rendszerint a klasztereken belül vannak és ezeknek egy finom szerkezetét mutatják. A fordított irány is előfordul, ilyenkor a klaszterek adnak információt a közösségekről. Azaz a legérdekesebb közösségek azok, amelyek elemei több klaszterhez tartoznak.

iv. A szociális gráfokban meggyőződtünk a gyenge kapcsolatok szerepéről Granovetter [20] és vizsgáltunk is néhány algoritmust. Az N^{++} által kapott közösségeken belül szinte kizárólag csak erős élek vannak, míg a gyenge élek a közösségek között vannak. A kisvilág gráfok másik típusánál az ún. technikai gráfoknál[4] ilyet nem tapasztaltunk. Adatainkat Hidalgó és társai [23] cikkéből vettük. (A CPM nem adott jó eredményt semmilyen k-ra, talán azért, mert túl érzékeny a mérési hibákra és a hiányzó adatokra.)

[4]A szociális gráfoknál az (x, y) és (x, z) élek megléte megnöveli az (y, z) él létezésének feltételes valószínűségét, míg a technikai gráfokban ilyenkor ez a valószínűség csökken.

v. Szociális gráfokban a csúcsoknak természetes attribútuma lehet az az időpont, amikor a csúcs csatlakozott a hálózathoz. Ez a sorrend nem mutatható ki, ha az egész hálózat klasztereit nézzük, de figyelemre méltó az egybeesés, ha csak egy kiválasztott csúcs szomszédságát tekintjük. Ebben az esetben a klaszterek néha jellemezhetők valamilyen idő intervallummal vagy térbeli korláttal. Megjegyzendő, hogy a közösségek átnyúlhatnak a klaszterek határain.

3.6. Dinamikus gráfok. Az alkalmazásokban fellépő gráfok függhetnek az időtől, így esetleg eldöntentő kérdés, melyik formájukat használjuk.[5] Az egyik alapvető feladat a közösségek nyomonkövetése, a változásának a leírása. Erre Palla és társai [30] és Bóta és társai [9] kísérelte meg. A megállapítások hasonló és eltérő elemeket egyaránt tartalmaznak; az utóbbinak sok forrása lehet. Az egyik, hogy míg a [30] kísérletei a CPM, a [9] szerzői az N^{++} algoritmust használták. Különböztek az adatbázisok, a [30] az ún. co-authorship gráfot és egy (amerikai) telefonhívási gráfot, míg a [9] egy banki tranzakciós gráfot és egy (magyar) telefonhívási gráfot elemzett. Végül a metodika is különbözött, a [30] szerzői egyszerű axiomatikus feltételekkel éltek a közösségekkel történhető elemi eseményekre (változatlan marad, eltűnik, kettéválik, egyesül, nő, zsugorodik), addig a [9] kísérletei megmutatták, hogy az esetek egy jelentős része nem fér bele ebbe a keretbe. Nyitott kérdéd, hogy az élek erőssége összefügg-e azzal, mennyire változó közösségekben húzónak az élek, lásd még az előző alfejezet iv. pontját.

3.7. Súlyozás. Súlyozott gráfokkal nehéz foglalkozni. Jóllehet az indirekt módszerek numerikus eredményei megbízhatóbbak, de ha ezeket kiterjesztjük súlyozott gráfokra, az eredmények még kevéssé ismertek Bóta [7].

Az alábbiakban az indirekt kiértékelés egy modelljét vázoljuk.[6] Az infekciós modellek a valódi gráfok alkalmazásának középpontjában állnak Boguña és Pastor-Satorras [4], de alkalmasat konstruálni nehéz. Fő szempontjai: (i) melyik modellt válasszuk, (ii) mik a lényeges változók és (iii) hogyan határozzuk meg a paraméterek értékét. Vizsgálataink a banki szféra két problémájára koncentrálódtak: 90 napot meghaladó nem fizetés, az ún. *hitel default* és általában a késedelmes fizetés, Csernenszky és társai [13, 14]. Hangsúlyozzuk, hogy bár a két probléma hasonló, mégis vannak köztük lényeges különbségek.

[5]Például a két egymás utáni hónapban a telefonhívásokból előállított gráfok élhalmaza csak kb. 30%-ban egyezik meg.

[6]Más megközelítéssel egy esettanulmányt vizsgálunk, amely bizonyította a hálózati modellek és a közösségek használhatóságát.

A fő hasonlóság a fenti két folyamatban, hogy mindkettő ragályos, azaz az üzleti partnereket is megfertőzheti. Mindazonáltal nagy gondossággal kell vizsgálni a jelenségeket, hiszen az üzleti nehézségek nem pusztán a környezetből adódhatnak, belső okaik is mindig vannak.[7] Tehát a feladatunk az, ha egy problémára, pl. a hitel default esetén, adottak egy-egy cég *apriori* valószínűségei, akkor becsüljük meg az *a posteriori* default valószínűségeket, amelyek egy fertőzési folyamat után értelmezettek. A valószínűségek különbségét tekinthetjük az adott problémában fellépő *hálózati hatásnak.* A probléma jellege miatt (azaz nincs felépülés, a fertőzés valószínűsége nem konstans az éleken) kizárjuk az epidemiológiában amúgy sikeres SIR vagy SIS modellek használatát. A célunknak legjobban a Független Kaszkád modell felel meg.

3.8. Független Kaszkád modell (IC). A Független Kaszkádról, vagy a megalkotói alapján Domingos-Richardson modellről lásd bővebben Domingos és Richardson, Kempe és társai [16, 24]. Megjegyezzük, hogy a modell egy ekvivalens változatát vizsgálta korábban Granovetter [21].

Adott egy G élsúlyozott gráf, ahol a (v, w) élhez a $p_{v,w}$ valószínűséget társítjuk. Az infekció az alábbi módon történik.

Az első lépésben a fertőzött csúcsok F_1 halmazát tekintjük aktívnak, azaz $F_1 = A_1$.

Általánosan a $w \in V(G) \setminus F_{i-1}$ csúcs $p = \prod_{v \in A_{i-1}} p_{v,w}$ valószínűséggel fertőződik meg az i-edik lépésben, és ekkor $w \in F_i$. A frissen fertőzött pontok a *rákövetkező lépésben* fertőzhetnek csupán, azaz $A_i = F_i \setminus F_{i-1}$. Ha valamely i-re $F_i = F_{i-1}$, akkor leáll a folyamat.

Megjegyezzük a pontok fertőzési valószínűségének kiszámítása nehéz probléma, jobbára szimulációkon alapul, lásd Kempe és társai, Csernenszky és társai [24, 13].

3.9. Súlyozás és optimalizálás. A megfelelő modellhez az IC modellt módosítanunk kell. Mivel az a posteriori fertőzési valószínűségeket úgyis szimulációkkal becsüljük, kézenfevő a szimuláció részévé tenni az apriori fertőzési valószínűségeket [14]. Ezzel a kezdeti fertőzés 0-1 értékei helyett teszőleges eloszlást használhatunk. Nagyobb problémát okoz a $p_{v,w}$ élfertőzési valószínűségek becslése, ezt az irányt a fenti cikk mellett az alábbiakban publikációkban kísérelték meg Goyal és társai, Saito és társai [19, 33]; sajnos alapvetően különböző feltevésekkel dolgozva.

[7] A gazdaság általános állapota figyelembe vehető egy fiktív ponttal, amely mindenkivel össze van kötve.

A megoldás a következőképpen történhet. A szokásos módon *tanuló* és *teszt* adatbázist veszünk fel. A $p_{v,w}$ valószínűségeket a tanulóhalmaz segítségével becsüljük, majd a teszthalmazzal mérjuk vissza. A másik probléma, hogy a $p_{v,w}$ valószínűségek becslése alulhatározott problémához vezet; itt azt feltételezzük, $p_{v,w}$ a v, w pontok és a (v, w) élhez tartozó attribútumoknak valamilyen (számunkra ismeretlen) függvénye. Ezt néhány paraméter segítségével fejezzük ki, majd a paramétereket optimalizáljuk, hogy minél jobban közelítse a tanulóhalmazban megadott tényleges fertőzési folyamatot. Végül meg kell választanunk a célfüggvényt, amely a becsléseink jóságát méri. A Bóta és társai [10] kutatásaiban ez a szokásos normákat jelenti, míg az alkalmazás jellege miatt a [14] az ún. *gain curve* megközelítést használta. Ebben a gráf pontjait a modell által (a teszthalmazon) számított fertőzési valószínűség szerinti *fordított sorrendbe* állítjuk. Legyenek ezek a valószínűségek $w_1 \geq, \ldots, \geq w_n$. Definiáljuk a *nyereség* (gain) függvényt a

$$\mathrm{gain}(x) = \frac{\sum_{i \leq x} w_i}{\sum_{i=1}^{n} w_i}$$

formulával és maximalizáljuk a

$$\int_{x=1}^{n} \mathrm{gain}(x) dx$$

értéket.

A $p_{v,w}$ élfertőzési valószínűségek az alább részletezendő attribútumokból lett felépítve. Szisztematikus kereséssel lettek kipróbálva a függvények[8] illetve a paraméterezésük. A végső aggregálása a traszformált értékeknek hasonlóan történt, míg a legjobb paraméter értékek keresése *grid search* által történt.[9]

3.10. Eredmények. Itt egyetlen kísérletet emelnénk ki a sok lehetséges modell közül. A részletes tanulmányt, amely az OTP KKV szektor adatbázisán alapult lásd [14]. A tranzakciós adatbázis 2008 augusztus és 2009 április (6 hónapos) időintervallumában rögzített adatin alapult a tranzakciós gráf, míg a fertőzési folyamat 2009 február és április (3 havi) adatait használta. A default események felvétele az alábbi két intervallumban történt: egy hosszabb 2009 május és 2010 április között (12 hónap), egy rövidebb pedig 2009 május és 2009 július között (3 hónap).

A következő tapasztalatok adódtak:

[8] Az alapfüggvények: lineáris, kvadratikus, logaritmus, exponenciális és szigmoid

[9] A tapasztalat szerint nagyobb feladatok megoldását adhatja a numerikus deriválás és a gradiens módszer megfelelő kombinációja, lásd [10].

I. A rövidebb (3 hónapos) default monitoron alapuló modellek jobban teljesítenek, mint a hosszabbon.

II. Az élek irányítása lényegés vevő-eladó formában kell felvenni, azaz ha x utal pénzt y-nak, akkor $(x, y) \in E(G)$.[10]

III. Indirekt élek. Ha van $x - z$ és $z - y$ tranzakció, de z nem ismert (pl. nem kliense az OTP-nek), a fertőzési modellben szerepet kaphat (x, y) élként elszámolva, ahol az attributumokra a IV/ii használandó.

IV. A lényegesnek bizonyult változók illetve rájuk vonatkozó tapasztalatok:

(i) A közösségi információ. (Adott él tartozik-e közösségbe?)

(ii) Az (x, y) él örökli az x változóit (de y-ét nem).

(iii) A relatív forgalom számít, azaz az élen küldött transzfer és a traszfer összegének hányadosa.

(iv) A kliens életkora. (Milyen öreg egy vállalat?)

(v) Viselkedés típusú változók. (queuing, overdraft stb.)

Mindazonáltal a legerősebb változók az (i) és (iii) pontban említettek.

A modellek által adott javítás az ún. *lift* segítségével értelmezhetők. A [14] szerint a defaultba eső kliensek megtalálásában a szektortól függően 3-4, egyes szektorokban (a legkockázatosabb ügyfelek esetén) 10-12-szeres lift adódik. A közösségi hatás erős, ha (x, y) egy közösségen belül futó él, akkor kb. háromszoros fertőzési valószínűséggel számolandó, a hasonló, de közösségen kívül futó élhez képest. Hasonló eredményekről számol be a [13] dolgozat.

4. Köszönetnyilvánítás

A kutatásokat az OTKA és a Magyar kormány és az Európai Unió "Social Renewal Operational Programme" keretében működő TÁMOP pályázatok támogatta. Az első szerzőt a TÁMOP-4.2.1/B-09/1/KONV-2010-0005, míg a második szerzőt az OTKA K76099 és futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

[1] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, T. Vicsek CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* **22**, 1021–1023 (2006).

[2] R. Albert and A. L. Barabási, Emergence of scaling in random networks. *Science* **286** (1999), no. 5439, 509–512.

[10] A modell irányítatlan élekkel is javítást hoz a hálózatot nem használó modellekhez képest; ezt egyfajta hálózati hatás okozza, hisz a gazdaság szereplői kölcsönös függésben vannak, illetve a hálózat a szektort is megragadja.

[3] R. Albert, A. L. Barabási Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74** 2002.

[4] M. Boguñá, R. Pastor-Satorras, A. Vespignani, Absence of epidemic threshold in scale-free networks with connectivity correlations. Preprint cond-mat/0208163 (2002).

[5] B. Bollobás, *Modern Graph Theory,* Springer, New York (1998).

[6] B. Bollobás and O. Riordan, Clique percolation. *Random Structures Algorithms* **35** (2009), no. 3, 294–322.

[7] A. Bóta, Applications of Overlapping Community Detection, $(CS)^2$ - Conference of PhD Students in Computer Science, Szeged, 2010.

[8] A. Bóta, L. Csizmadia and A. Pluhár, Community detection and its use in Real Graphs. *Proceedings of the 13th International Multiconference INFORMATION SOCIETY - IS* 2010 Volume A 393–396.

[9] A. Bóta, M. Krész and A. Pluhár, Dynamic Communities and their Detection. *Acta Cybernetica* **20** (2011) 35–52.

[10] A. Bóta, M. Krész and A. Pluhár, Systematic learning of edge probabilities in the Domingos-Richardson model. Int. J. Complex Systems in Science Volume **1(2)** (2011) 115–118.

[11] A. Cami, N. Deo, Techniques for analyzing dynamic random graph models of web-like networks: An overview. *Networks* **51** (2008) No. 4, 211–255.

[12] Luciano da Fontoura Costa, Hub-Based Community Finding. arXiv:cond-mat/0405022 v1 3 May 2004.

[13] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár and T. Tóth, The use of infection models in accounting and crediting. Challenges for Analysis of the Economy, the Businesses, and Social Progress, Szeged, 2009.

[14] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár and T. Tóth, Parameter Optimization of Infection Models. $(CS)^2$ - Conference of PhD Students in Computer Science, Szeged, 2010.

[15] L. Csizmadia, Recognizing communities in social graphs, MSc thesis, University of Szeged, 2003.

[16] P. Domingos, M. Richardson, Mining the Network Value of Costumers. *7th Intl. Conf. on Knowledge Discovery and Data Mining,* 2001.

[17] T. S. Evans and R. Lambiote, Edge Partitions and Overlapping Communities in Complex Networks. arXiv:0912.4389v1, 2009.

[18] S. Fortunato, Community Detection in graphs. arXiv:0906.0612

[19] A. Goyal, F. Bonchi and L.V.S. Lakshmanan, Learning influence probabilities in social networks. *WSDM '10 Proceedings of the third ACM international conference on Web search and data mining ACM* New York, NY, USA (2010) doi: 10.1145/1718487.1718518

[20] M. Granovetter, The Strength of Weak Ties. *American Journal of Sociology* **78(6)** 1360–1380, 1973.

[21] M. Granovetter, Threshold models of collective behavior. *American Journal of Sociology* **83(6)** 1420–1443, 1978.

[22] E. Griechisch, Clustering and community finding methods in graphs, MSc thesis, University of Szeged, 2010.

[23] C. A. Hidalgo, B. Klinger, A. L. Barabási and R. Hausmann, The Product Space Conditions the Development of Nations. *Science* (2007) 317: 482–487.

[24] D. Kempe, J. Kleinberg and E. Tardos, Maximizing the Spread of Influence though a Social Network. *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining,* 2003.

[25] J. Kleinberg, An Impossibility Theorem for Clustering. *Advances in Neural Information Processing Systems (NIPS)* 15, 2002.

[26] I. A. Kovács, R. Palotai, M. S. Szalay and P. Csermely, (2010) Community Landscapes: An Integrative Approach to Determine Overlapping Network Module Hierarchy, Identify Key Nodes and Predict Network Dynamics. *PLoS ONE* 5(9): e12528. doi:10.1371/journal.pone.0012528

[27] T. Népusz, A. Petróczi, L. Négyessy and F. Bazsó, Fuzzy communities and the concept of bridgeness in complex networks. arXiv:0707.1646v3, 2007.

[28] M.E.J. Newman, The structure and function of complex networks. Preprint cond-mat/0303516 (2003).

[29] G. Palla, I. Derényi, I. Farkas and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814 (2005).

[30] G. Palla, A.-L. Barabási and T. Vicsek, Quantifying social group evolution. *Nature* **446**, 664–667 (2007).

[31] A. Pluhár, A telefonos logfile-on alapuló ismeretségi gráfok klasztereiről. Research Report 2001.

[32] A. Pluhár, Ismeretségi gráfok közösségeinek meghatározása gyors algoritmusokkal. Research Report 2002.

[33] K. Saito, R. Nakano and M. Kimura, Prediction of Information Diffusion Probabilities for Independent Cascade Model. *Knowledge-Based Intelligent Information and Engineering Systems* Lecture Notes in Computer Science, 2008, Volume 5179/2008, 67–75, DOI: 10.1007/978-3-540-85567-5_9

[34] D. Stark, The vertex degree distribution of random intersection graphs. *Random Structures and Algorithms* **24(3)**, (2004), 249–258.

[35] W. W. Zachary, An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33** (1977) 452–473.

Department of Computer Science, University of Szeged, Hungary
E-mail address: bartalos@inf.u-szeged.hu, pluhar@inf.u-szeged.hu