NeMa: Fast Graph Search with Label Similarity (NeMa: Gyors gráfkeresés címke hasonlóság alapján)

Arijit Khan, Yinghui Wu, Charu C. Aggarwal, Xifeng Yan

Pillinger János, Németh Bence, Bereczki Gábor

November 26, 2013

Bevezetés

- Az internet elterjedésével adatok forrása drasztikusan megnőtt, beleértve a világhálót, szociális hálózatokat, tudás gráfokat
- Ezáltal egyre gyakrabban szükséges olyan gráfokban keresni, amelyek címkézett, heterogén összetevők hálózataként vannak reprezentálva (csúcsok - címkézett entitások, élek - köztük lévő kapcsolatok)
- Ehhez azonosítanunk kell egy adott keresett (query) gráf illeszkedéseit egy (tipikusan hatalmas) hálózatban (ezt nevezzük célgráfnak)
- Probléma: A célgráfban lévő zaj és fixált séma hiánya miatt, a keresett gráf jelentősen különbözhet a célgráfban lévő illeszkedésektől (struktúra / csúcscímkézés)

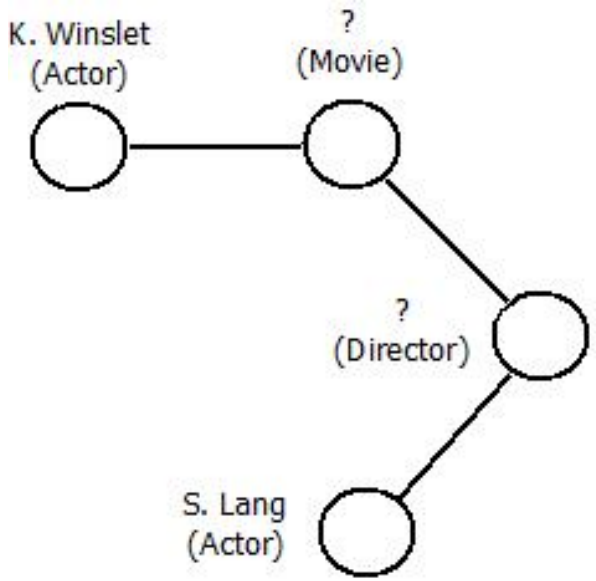
A cikkben

- Bevezetnek egy újszerű részgráfillesztési költségmetrikát, ami összegzi az egyedi csúcsok illesztésének költségét és egyesíti a strukturális és csúcscímkézési hasonlóságokat
- Ezen metrikára alapozva megfogalmaznak egy minimum költségű részgráfillesztő problémát: a keresett gráf (legjobb-k) minimum költségű illesztésének megtalálása a célgráfban (NP-nehéz)
- Heurisztikus algoritmust ajánlanak egy következtetési modellre alapozva
- Valamint optimalizálási technikákat is kínálnak az eljárás hatékonyságának javítására

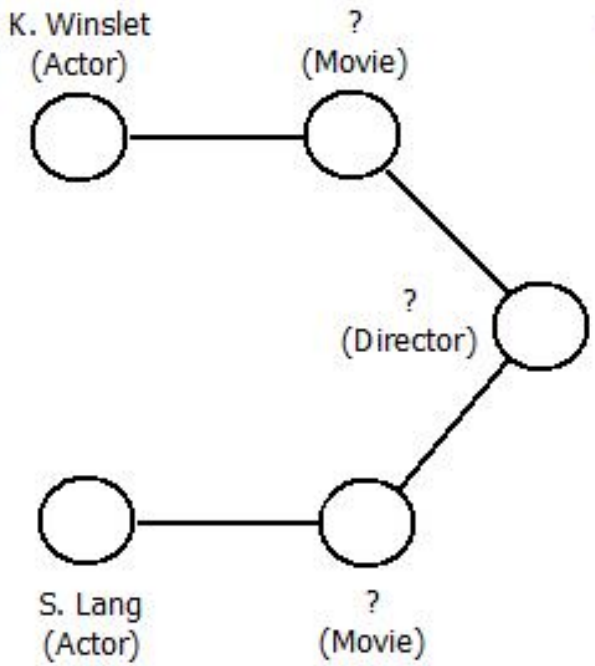
Példa

- Tegyük fel, hogy a felhasználó egy olyan filmet szeretne megtalálni, amelynek szereplője 'Kate Winslet' és ugyanaz a rendező rendezte aki egy 'Stephen Lang' által játszott filmet is
- Tételezzük fel továbbá, hogy a séma és pontos címkéi az entitásoknak ismeretlenek a célhálózatban
- Mindezek ellenére a felhasználó ekkor is találhat a kereséshez néhány ésszerű gráfreprezentációt
- Magától értetődő hogy az ilyen grafikus reprezentációk nem lesznek egyediek, hiszen egy keresésre szolgáló ábrát módunkban áll többféleképpen felrajzol

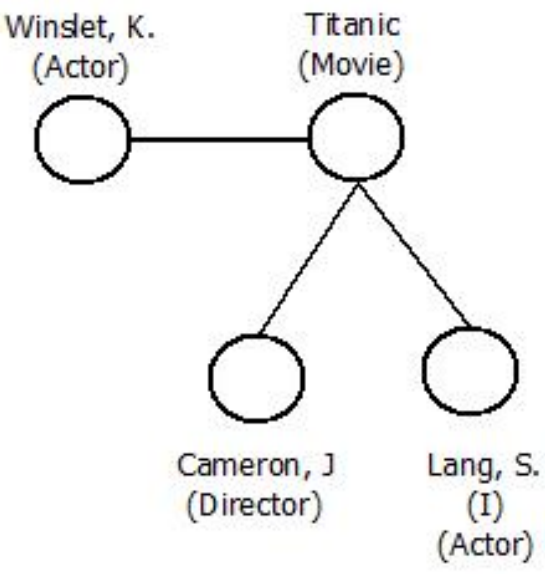
Lehetséges gráfreprezentációk



a.) Keresett gráf 1



b.) Keresett gráf 2



c.) Top-1 illesztés

Az első két ábra a feladat 1-1 lehetséges keresett gráfját reprezentálja, míg a harmadik az adatok tényleges struktúráját. Az első két ábra lánctopológiájú, míg a harmadik csillagtopológiájú gráfot ábrázol.

Meglévő technikák vizsgálata

- Tradicionális gráfkereső modellek általában részgráfizomorfizmus és kiterjesztései szempontjából vannak definiálva
- Ezek csak azon részgráfok azonosítására alkalmasak, amelyek pontosan, vagy megközelítőleg izomorfak a keresett gráfokkal
- Használhatjuk továbbá lekérdező modellek és nyelvek széles körét: SPARQL, XDD, az RDF és XML adatokhoz, de ezek megkívánják az általános séma meglétét
- Problémát jelent tehát, hogy a valós gráfok komplexek, zajosak, és gyakran hiányolják az általánosított sémákat
- Így a tradicionális gráfkereső technikák nem képesek jó minőségű illesztéseket találni

A NESS-ről

- Manapság a NESS-t kínálják részgráfillesztésre, amely figyelembe veszi a csúcsok közelségét, de egy szigorú csúcscímkeillesztést alkalmaz
- Ez első lépésben egy szűrő fázist alkalmaz, amelyben a kevésbé ígéretes csúcsjelölteket iteratívan szortírozza. Ennek kimenetele egy limitált számú végső jelölt mindegyik keresett csúcs számára
- Ezután az algoritmus ezen végső jelölteket felhasználva megvizsgálja az összes lehetséges gráfilleszkedést, hogy megtalálja a legjobb-k gráfillesztést
- Módosítható úgy, hogy figyelembe vegye a csúcscímke különbségeket is, azonban ez a módosítás drasztikusan csökkenti a szűrő fázis hatékonyságát

Alapfogalmak

- **Célgráf:** A heterogén hálózatokat reprezentáló célgráfok definiálhatók címkézett, irányítatlan gráfként $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{L})$, amiben a csúcsok halmaza $\mathbf{V}$, az élek halmaza $\mathbf{E}$, és a címkéző függvény $\mathbf{L}$, és
  - minden $\mathbf{V}$-beli $\mathbf{u}$ célcsúcs egy hálózatbeli entitást reprezentál
  - minden $\mathbf{E}$-beli $\mathbf{e}$ él két entitás közti kapcsolatot fejez ki
  - $\mathbf{L}$ egy függvény amely minden $\mathbf{u}$ csúcshoz egy $\mathbf{L}(\mathbf{u})$ véges ábécéjű címkét rendel

  Tehát a gyakorlatban a csúcscímkék reprezentálhatják az entitások tulajdonságait, például név, érték, stb.

Keresett gráf és címkekülönbség függvény

- **Keresett gráf:** A keresett gráf $\mathbf{Q} = (\mathbf{V_Q}, \mathbf{E_Q}, \mathbf{L_Q})$ egy irányítatlan, címkézett gráf, amiben $\mathbf{V_Q}$ a keresett csúcsok halmaza, $\mathbf{E_Q}$ az élek halmaza, és $\mathbf{L_Q}$ egy címkéző függvény, ami minden $\mathbf{V_Q}$-beli $\mathbf{v}$ keresett csúcshoz rendel egy $\mathbf{L_Q(v)}$ véges ábécéjű címkét.

- **Címkekülönbség függvény:** Addot tehát egy $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{L})$ célgráf és egy $\mathbf{Q} = (\mathbf{V_Q}, \mathbf{E_Q}, \mathbf{L_Q})$ keresett gráf. A $\mathbf{V}$-beli $\mathbf{u}$ csúcs egy jelöltje a $\mathbf{V_Q}$-beli $\mathbf{v}$ keresett csúcsnak, ha a különbség a címkék között (értsd $L(u)$ és $L_Q(v)$) egy adott $\mathbf{\Delta_L}$ címkekülönbség függvény által meghatározva kevesebb vagy egyenlő, mint az előre megszabott küszöb. A $\mathbf{v}$ csúcs jelölthalmazát $\mathbf{M(v)}$-vel jelöljük. A részgráfillesztés egy 'many-to-one' függvény: $\mathbf{V_Q} \rightarrow \mathbf{V}$, ami minden $\mathbf{V_Q}$-beli $\mathbf{v}$ keresett csúcshoz $\mathbf{M(v)}$-beli eredményt ad.

Részgráfillesztési költség

- Több illesztőfüggvény is megadható (a cikk írói a Jaccard-hasonlóság mérést használták)
- A gráfhasonlósági metrikának meg kell őriznie a keresett gráfban lévő csúcsok közti közelséget, miközben az illesztett csúcsok címkéjének is hasonlónak kell lenniük
- A NEMA esetében ez a függvény összeadja a keresett csúcs jelölt csúcsra való illesztésének költségeit, így ragadva meg két csúcs címkéit, és szomszédságstruktúrái közötti különbséget

Szomszédságvektorizálás

Szomszédságvektorizálás: Legyen adott egy **u** csúcs a **G** célgráfban, *i* szomszédságát a szomszédságvektorral reprezentáljuk $\mathbf{R_G(u)} = \{< \mathbf{u'}, \mathbf{P_G(u, u')} >\}$, ahol u' egy olyan csúcs amely h távolságon belül van az u csúcstól, és $\mathbf{P_G(u, u')}$ jelöli **G**-ben vett közelségét a két csúcsnak.

$$P_G(u, u') = \begin{cases} \alpha^{d(u,u')} & \text{ha} \quad d(u, u') \leq h; \\ 0 & \text{egyébként.} \end{cases}$$

Itt a $\mathbf{d(u, u')}$ a távolság **u** és **u'** csúcsok között. A terjedési paraméter (α) 0 és 1 közötti, és $h > 0$ a távolság (azaz a sugara) a verifikáció szomszédságának. Az **u** csúcs szomszédainak vektora magában foglalja a közelségek információját az **u** csúcstól a h távolságra lévő szomszédcsúcsokig. Gyakran elegendő kis értéket választani **h** számára (pl.: $h = 2$), mivel két entitás közötti kapcsolat irrelevánssá válik ahogy a szociális távolság köztük nő.

Illesztési költség modellezése

Illesztési kölség modellezése: A szomszédok vektorára alapozva, most tovább lépünk a keresett csúcs és célcsúcs szomszédainak illesztési költségének modellezésére. Jelöljük a **v** csúcs **h** távolságra lévő szomszédcsúcsainak halmazát **N(v)**-vel. Ha adott a ϕ illesztő függvény, a **v** és **u** $= \phi(\mathbf{v})$ közti szomszédillesztési költséget

$$N_\phi(v,u) = \frac{\sum_{v' \in \mathbb{N}(v)} \Delta_+(P_Q(v,v'), P_G(u,\phi(v')))}{\sum_{v' \in \mathbb{N}(v)} P_Q(v,v')}$$

ahol a $\Delta_+(x,y)$ a következőképpen van definiálva

$$\Delta_+(x,y) = \begin{cases} x-y, & \text{ha} \quad x > y \\ 0 & \text{egyébként.} \end{cases}$$

Illesztési költség modellezése

Tehát a $\mathbf{N_\phi(v, u)}$ felméri a v és u szomszédvektorainak az illesztési költségét. A $\mathbf{\Delta_+}$ elkerüli azon esetek büntetését, amikor két csúcs közelebb van a célgráfban, mint a hozzájuk tartozó csúcsok a keresett gráfban. A különböző **csúcsok illesztési költségét** az illesztő függvény szerint úgy definiáljuk, hogy vesszük a **lineáris kombináció**ját a **címkekülönbség függvénynek** és a **szomszédságillesztési költségfüggvénynek**.

$$F_\phi(v, u) = \lambda \cdot \Delta_L(L_Q(v), L(u)) + (1 - \lambda) \cdot N_\phi(v, u)$$

ahol

$$u = \phi(v)$$

Illesztési költség modellezése

A csúcsillesztési költség kombinálja a címkeillesztési költséget és a szomszédságillesztési költséget. Most pedig definiáljuk a részgráfillesztő költségfüggvényt. Adva van egy ϕ illesztés a V_Q-beli v keresett csúcsról a V-beli $\phi(v)$ célcsúcsokra, ekkor a részgráfillesztés költségfüggvény definíciója:

$$C(\phi) = \sum_{v \in V_Q} F_\phi(v, \phi(v))$$

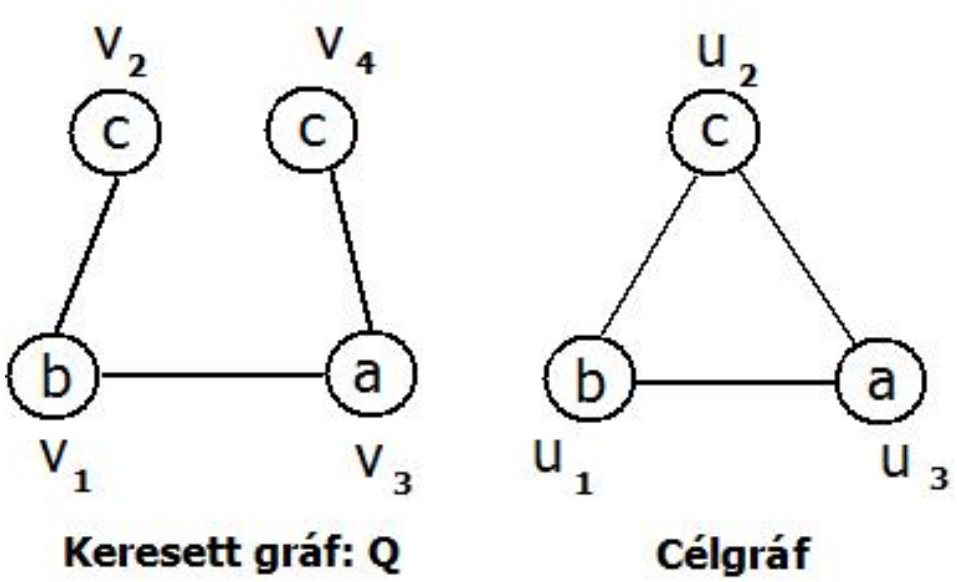
Látható hogy a $C(\phi)$ az illesztés költsége ϕ -nek a Q keresett gráf és a G célgráf között, és a probléma az hogy találjunk egy ϕ illesztő függvényt, amivel $C(\phi)$ minimum értéket vesz fel.

Jelölések

- Keresett gráf: $Q(V_Q, E_Q, L_Q)$
- Célgráf: $G(V, E, L)$
- Részgráfillesztő függvény: $\phi : V_Q \rightarrow V$
- Címkekülönbség függvény: Δ_L
- V csúcs jelölthalmaza: $\mathbb{M}(v)$
- U csúcs szomszédságvektora: $\mathbb{R}_G(u)$
- Szomszédságillesztés költsége u és v között: $N_\phi(v, u)$
- Egyéni csúcsillesztés költsége u és v között: $F_\phi(v, u)$
- Részgráfillesztési költségfüggvény: $C(\phi)$

Példa

Vegyünk egy Q keresett gráfot, egy G célgráfot, és egy ϕ részgráfillesztő függvényt, ahol $\phi(v_1) = u_1$, $\phi(v_2) = \phi(v_4) = u_2$, és $\phi(v_3) = u_3$. Legyen $h = 1$ és $\alpha = 0.5$, a és határozzuk meg Q és G szomszédvektorait, ebből a szomszédillesztési költséget, majd az egyéni csúcsillesztés költségét u és v között, végül pedig részgráfillesztési költségfüggvényt!



$R_Q = ?,\ R_G = ? \longrightarrow N_\phi \longrightarrow F_\phi \longrightarrow C(\phi)$

Költségfüggvény tulajdonságai

Tulajdonság 1: Ha a Q keresett gráf részgráfizomorfikus (struktúra és csúcscímkék egyenlősége szempontjából) a G célgráfra, akkor létezik minimum költségű illesztő függvény, melyre $C(\phi) = 0$. Az olyan $C(\phi) = 0$ illesztéseket, amelyekre teljesül hogy nem izomorfikusak Q-ra, "hamis pontos illesztés"-ként tekintünk.
Tulajdonság 2: Ha a ϕ-vel jelölt illesztés nem izomorfikus a Q keresett gráfra, és ϕ egy 'one-to-one' típusú függvény, akkor $C(\phi) > 0$.

Eredmények

Adott tehát G célgráf, Q keresett gráf, és a címkézési zajküszöb, találj minimum költségű illesztést,

$$argmin_{\phi}(C(\phi))$$

$$\Delta_L(L_Q(v), L(u)) \leq \varepsilon, \forall v \in V_Q, u = \phi(v)$$

A probléma megfogalmazásunk tehát egy optimális illesztést azonosít csúcscímke különbségek és csúcspár távolság minimalizálással.
Állítás 1.: Adott G célhálózat, Q keresett gráf. Annak eldöntése, hogy létezik-e ϕ illesztés NEMA-ban $C(\phi) = 0$ részgráfillesztési költséggel, egy NP-teljes probléma.
Állítás 2.: A minimum költségű részgráfillesztés APX-nehéz (közelíteni is nehéz).

Iteratív következtető algoritmus - NemaInfer

Állítás: A Max-sum következtetési probléma hasonló a minimum költségű algráf illesztés problémához, amely megköveteli a $C(\phi)$ részgráfillesztési költségfüggvény minimalizálását.

Max-sum következtetés: grafikus modelleknél esetében az együttes valószínűségi sűrűségfüggvény ($p(X)$) az $X = x_1, x_2, \ldots, x_M$ változókra megadható a következő formula segítségével. $p(X) = \prod_i f_i(X_i)$, ahol minden $X_i \subseteq X$.

Alternatív megfogalmazása: $\log p(X) = \sum_i \log f_i(X_i)$. A max-sum következtetési probléma, hogy megtaláljuk azon $x_1, x_2, \ldots, x_M$ értékeket, ahol a $p(X)$ a maximum értékét veszi fel.

Iteratív következtető algoritmus ? NemaInfer

Bemenet: $G(V, E, L)$ célgráf, $Q(V_Q, E_Q, L_Q)$ keresett gráf.
Kimenet: A Q min költségű illesztése G-re
for all csúcs $v \in V_Q$ **do**
  kiszámolni $\mathbb{M}(v)$-t;
  i := 0; flag := true;
  Iteratív módon meghatározza au U_i-k értékeit (1-es képlet);
  while flag **do**
    i := i + 1;
    for all $v \in V_Q$ **do**
      for all $u \in \mathbb{M}(v)$ **do**
        kiszámítjuk $U_i(v, u)$-t a 2. képlet segítségével:
        nyomon követjük a jelenlegi helyezetét a szomszédoknak $v' \in \mathbb{N}(v)$;
      end for
      nyomon követjük az összes keresett csúcs optimális illesztését $O_i(v)$ az 3-es képlettel;
    end for
    if ezt addig ismételjük amíg egy fixponthoz nem érünk, v kielégíti $O_i(v) = O_{i-1}(v)$-et **then**
      flag = false;
    end if
  end while
  előállítjuk ϕ minden $v \in V$-re az F_ϕ-k és U_i-k segítségével
  return ϕ
end for

Képletek az algoritmushoz

$$U_0(v,u) = \min_{\phi:\phi(v)=u} F_\phi(v,u) \tag{1}$$

$$U_i(v,u) = \min_{\phi:\phi(v)=u} [F_\phi(v,u) + \sum_{v' \in \mathbb{N}(v)} U_{i-1}(v',u')] \tag{2}$$

$$O_i(v) = \operatorname*{argmin}_{u \in \mathbb{M}(v)} U_i(v,u); i \geq 0 \tag{3}$$

Hatékonyság

Végezetül pedig vessük össze más Kulcsszavas keresés és gráfkereső eljárásokkal.

	Nema	BLINKS	IsoRank	SAGA
Precision	0.91	0.52	0.63	0.75
Recall	0.91	0.52	0.63	0.75

Ahol a precízió (Precission) a helyesen felfedezett gráfillesztések és az összes felfedezett gráfillesztések arányaként van definiálva.
A visszahívás (Recall) pedig a helyesen felfedezett gráfillesztések és az összes helyes gráfillesztés arányaként van mérve.

Köszönöm a megtisztelő figyelmet!

$$U_i(v,u) = \Delta_L(L_Q(v), L(u)) + \sum_{v' \in \mathbb{N}(v)} W_i(v,u,v')$$

ahol

$$W_i(v,u,v') = \min_{\phi:\phi(v)=u}[\beta(v) \cdot \Delta_+(P_Q(v,v'), P_G(u,\phi(v'))) + U_{i-1}(v',\phi(v'))]$$

ahol

$$\beta(v) = [\sum_{v' \in \mathbb{N}(v)} P_Q(v,v')]^{-1}$$

$$\phi(v) = \operatorname*{argmin}_{u \in \mathbb{M}(v)} U_{i'}(v,u) \tag{4}$$

$$\phi_p = \arg \min_{\phi:\phi(v)=\phi(v)} [F_\phi(v,\phi(v)) + \sum_{v' \in \mathbb{N}(v)} U_{i'-1}(v',\phi(v'))] \tag{5}$$

ahol

$$\phi(v') = \phi_p(v')$$