hyväksymispäivä arvosana

arvostelija

Aine

Ola Länsman

Helsinki 12.3.2017

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

1 Johdanto

Käytännön kokeilla sosiaalisista verkoista muodostettu Pieni maailma-ilmiö on vaikuttanut myös tietojenkäsittelytieteeseen. Ilmiötä on yleistetty useampiin ulottuvuuksiin sekä sen tärkeimpiä ominasuuksia kuten klusteroitumista tutkittu. Ilmiön perusteella on tehty useita polunetsintä-algoritmeja jotka käyttävät ainoastaan lokaalia tietoa verkon solmuista ja kaarista.

Tutkielma pyrkii selittämään Pieni maailma-ilmiön verkkoteoriaa ja muutaman yleisen polunetsintä-algoritmin. Näiden jälkeen kiinnostuksen kohteeksi nousevat käytännön sovellukset joissa näitä käytetään.

2 Pieni maailma

2.1 Määritelmä

Pieni maailma-ilmiö tarkoittaa seuraavan ehdon toteumista verkossa:

Lemma 1 *Yksittäiset solmut voivat lähettää viestin verkon kaaria pitkin muihin verkon solmuihin lyhyitä polkuja pitkin käyttämällä ainoastaan paikallista tietoa.*

Paikallisella tiedolla tarkoitetaan globaalin tiedon puuttumisella. Jokaisessa solmussa valitaan viestin seuraava vastaanottaja käyttäen hyväksi ainoastaan tietoa viestiä lähettävän solmun kaarista. [DHLS06]

Pieni maailma-verkot (PM) yleensä täyttävät myös seuraavat ehdot:

1. *ne ovat harvoja*[BBS11]: 1 täyttyy implisiittisesti, mikäli lähtösolmusta löytyy kaari kohteeseen. Tälläiset verkot eivät ole mielenkiintoisia ilmiön kannalta.
2. *solmujen välillä esiintyy lyhyitä polkuja* [BBS11]: 1 aiheuttaa tämän suoraan.
3. *ne ovat ryhmittyneitä ja niillä on pieni halkaisija* [BBS11]: Intuitiivisesti ehto 3 tarkoittaa, että mikäli solmu u ja v ovat lähellä toisiaan niin niiden välillä on todennäköisesti kaari. Yleisimmissä (ahneissa) polunetsintä-algoritmeissa viesti lähetetään aina mahdollisimman lähelle kohdetta ehdosta 3 johtuen.

2.2 Kuinka muodostetaan pieni maailma

2.2.1 Kleinbergin malli

PM-verkkoja voidaan muodostaa monella tavalla. Aloitamme muodostamalla mallin, joka tehtiin kuvaamaan verkkoa ihmisten sosiaalisista suhteista.

Mallinnamme ihmisiä solmuina. Tämä joukko solmuja muodostaa $n \times n$ ruudukon, missä

$$V = \{(i, j) : i \in \{1, 2, \ldots, n\}, j \in \{1, 2, \ldots, n\}\} .$$

Olkoon ruudukkoetäisyys $d((i, j), (k, l)) = |k - i| + |l - j|$. Solmulla u on *paikallinen kontakti* solmun v kanssa, jos $d(u, v) \leq p$ jollain vakiolla $p \geq 1$. Solmulla u on myös vakiomäärä, $q \geq 0$, *etäkontakteja.* Etäkontaktit muodostetaan satunnaisesti todennäköisyysfunktiolla, joka riippuu vakiosta $r \geq 0$ ja etäisyydestä $d(u, v)$. Tarkemmin, etäkontakti muodostuu solmusta u solmuun v todennäköisyydellä $[d(u, u)]^{-r}$. [Kle00]

Tämä tapa muodostaa PM-verkko voidaan tulkita myös geometrisesti. Solmulla u on kaari jokaiseen tarpeeksi lähellä olevaan solmuun. Näiden yhteyksien lisäksi solmulla u on kaaria kauempana ruudukossa. Jos vakio $r = 0$, niin solmujen etäkontaktit ovat jakautuneet tasaisesti ruudukolle. Vakion r kasvaessa etäkontaktit ovat lähempänä ja lähempänä solmua itseään. [Kle00]

Tässä mallissa ovat *etäkontaktit* mielenkiintoisimpia tarkastelun kohteita. Verkon *navigoitavuuteen* vaikuttaa, kuinka tasaisesti etäkontaktit ovat jakautuneet verkkoon. Jos $r = 0$ eli etäkontaktit olisivat jakautuneet tasaisesti verkkoon, niin ahne polunetsintä-algoritmi ei tuottaisi lyhyitä polkuja luotettavasti. Vaikka algoritmi löytäisi solmulta x kaaren solmuun y joka on lähellä kohdetta z, niin solmun y todennäköisyys omata etäkontakti kohteeseen z ei olisi kasvanut. Vakion r ollessa liian suuri, hyppyjen määrä olisi myös suuri sillä viesti ei pääsisi kulkemaan tarpeeksi pitkälle etäkontaktien avullakaan.

Huomaamme, että etäkontaktien ei tarvitse olla satunnaisesti tuotettuja luodaksemme navigoitava PM-verkko. Etäkontaktien satunnaisuutta vähentämällä verkosta muodostuu ryhmittyneempi, joka edesauttaa mm. vertaisverkkojen virheenkestävyyttä. [CG06a]. Rajoitamme etäkontaktien valitsemisen satunnaisuutta rajoittamalla solmun $u = (u_1, u_2)$ mahdollisiksi etäkontakteiksi vain solmut $v = (v_1, v_2)$ joille joko $u_1 = v_1$ tai $u_2 = v_2$. Tällöin solmun etäkontatkit sijaitsevat samalla suoralla solmun itsensä kanssa ja etäkontaktien satunnaisuus pienenee. Artikkelissa (inproceeding?) [CG06a] on todistettu, että etäkontaktien muodostamista rajoittamisesta huolimatta verkon navigoitavuus säilyy. Edellisen ehdon lisäksi voitaisiin myös määritellä muita ehtoja, jotka koskevat vain osaa solmuista (*yhteisöt*). Myös tällöin verkolla säilyisivät samat ominaisuudet. [CG06a]

3 Polunetsintä

3.1 Strategiat

3.1.1 Ahne reititys

Ensimmäisenä tutkimme normaalia ahnetta algoritmia lyhyen polun etsintään. Ahneissa algoritmeissa viestiä kuljettava solmu lähettää viestin aina lähimpänä kohdetta olevalle naapurilleen. Esimerkiksi Kleinbergin esittämässä ahneessa algoritmissa viestiä kuljettava solmu tietää

1. kaikkien solmujen paikalliset kontaktit,
2. kohteen y sijainti ruudukossa
3. ja kaikkien viestiä kuljettaneiden solmujen etäkontaktit ja sijainnit. [Kle00]

.

Luvussa 2.2.1 esitetylle PM-verkolle voidaan muodostaa ahne algoritmi, jonka keskimääräinen *hyppyjen* (monellako solmulla viesti on käynyt) määrä on $\mathcal{O}(log^2 n)$. [Kle00] Käytämme tätä suuretta myöhemmin vertailua varten.

3.1.2 Epäsuora ahne reititys

Epäsuora ahne reititys toimii kuten ahne reititys. Poikkeuksena, viestiä kuljettavalla solmulla u on tiedossa myös solmun v etäkontaktit jos solmulle pätee $d\,(u, v) \leq q$ jollain etäisyysfunktiolla d ja vakiolla q. Tällöin viestiä kuljettavalla solmulla on mahdollisuus lähettää viesti jonkin itseään lähellä olevan solmun kautta.

3.1.3 Naapurien naapurit

Ahneella naapurien naapurit-algoritmi toimii samalla periaatte kuin epäsuora ahne reititys. Lähellä olevien solmujen sijaan viestiä kuljettavalla solmulla on tiedossa omien naapureidensa kontaktit.

3.2 NN-Ahne

Esitämme erään Naapurien Naapurit-ahneet algoritmin. Kiinnitämme huomiota muodostetun polun pituuteen, jonka merkitys on suurempi mm. vertaisverkkosovelluksissa. Erityisesti vertaamme sitä luvussa 3.1.1 mainittuun algoritmiin, jonka keskimääräinen hyppyjen määrä yhdessä PM-verkossa on $\mathcal{O}(log^2 n)$.

Algoritmi viestin lähettämisestä solmuun u toiminta kulkee seuraavalla tavalla:

1. Oletetaan viestin olevan solmussa $u \neq t$. Olkoon solmut $w_1, w_2, \ldots, w_k$, solmun u naapureita.
2. Kullekin i olkoon solmut $z_{i_1}, z_{i_3}, \ldots, z_{i_k}$ solmun w_i naapureita.
3. Oletetaan solmun z_{i_j} olevan tästä joukosta lähimpänä kohdetta t.
4. Lähetetään viesti solmuun z_{i_j} solmun u_j kautta.

.

4 Käytännön sovellukset

4.1 Peer-to-peer verkot

4.2 Luonnollisen kielen tiivistäminen

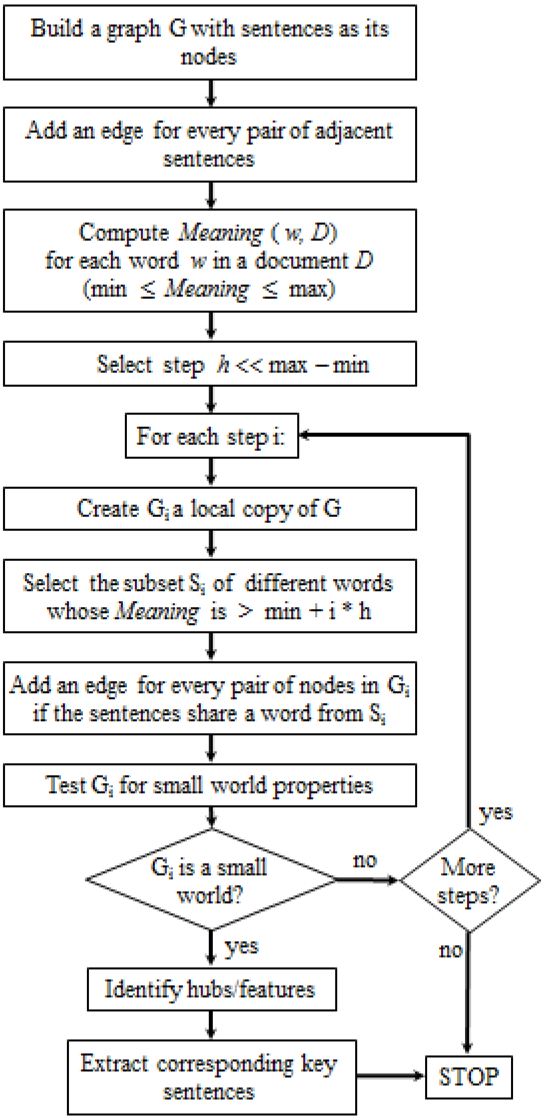
Luonnollisten kielten tekstien tiivistäminen on hyödyllistä nykymaailmassa elektronisesti käsillä olevan tiedon kasvaessa. Tekstiä voi tiivistää valikoiden ydinkohdat tai esittäen nämä lyhyesti uusin sanoin. Seuraavaksi esittelemme automaattisen valikoivan tiivistämiskeinon, joka käyttää hyväkseen pienten maailmojen topologiaa. Rakennamme tekstin virkkeiden verkon ja poimimme virkkeistä ne, jotka edesauttavat verkkoa eniten olemaan pieni maailma.

Määrittelemme verkon $G = (V, E)$, jossa pisteet V ovat lauseita ja kaaret E kuvaavat virkkeiden välisiä suhteita. Virkkeellä L on lähikontakti virkkeen L' kanssa, jos virkkeet L ja L' ovat peräkkäisiä. Etäkontaktien muodostamiseen tarvitsemme keinon määrittää virkkeiden yhteyden toisiinsa. Tätä varten rakennamme tekstin tärkeimmistä sanoista joukon $MeaningfulSet(e)$. Etäkontakti kahden virkkeen välille muodostuu vain, jos kummassakin virkkeessä esiintyy jokin joukon $MeaningfulSet(e)$ sanoista.

Joukon MeaningfulSet(e) sanojen määrä suhteessa joukkoon kaikista tekstissä esiintyvistä sanoista vaikuttaa verkon G kaarien määrän ja täten myös tiivistelmän pituuteen ja olennaisuuteen. Jos joukko $MeaningfulSet(e)$ on liian suuri, verkko G ei näytä enää pieneltä maailmalta vaan sattumanvaraisesti muodostetulta verkolta. Kuitenkin joukon $MeaningfulSet(e)$ ollessa liian pieni verkko G näyttää molempiin suuntiin linkitetyltä listalta josta löytyy muutama poikkeus. Tiivistysmenetelmän toimintaperiaatteen kannalta tällöin on suuresti merkitystä, miten tämä joukko valitaan. Tätä menetelmää emme esitä tässä tutkielmassa.

[BBS11]

Kuva 1 (Automaattinen tekstin tiivistäminen)



[]

Lähteet

BBS11 Balinsky, H., Balinsky, A. ja Simske, S. J., Automatic text summarization and small-world networks. *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, New York, NY, USA, 2011, ACM, sivut 175–184, URL `http://doi.acm.org/10.1145/2034691.2034731`.

CG06a Cordasco, G. ja Gargano, L., How much independent should individual contacts be to form a small–world? *Proceedings of the 17th International Conference on Algorithms and Computation*, ISAAC'06, Berlin, Heidelberg, 2006, Springer-Verlag, sivut 328–338, URL `http://dx.doi.org.libproxy.helsinki.fi/10.1007/11940128_34`.

CG06b Cordasco, G. ja Gargano, L., How much independent should individual contacts be to form a small–world? *Proceedings of the 17th International Conference on Algorithms and Computation*, ISAAC'06, Berlin, Heidelberg, 2006, Springer-Verlag, sivut 328–338, URL http://dx.doi.org.libproxy.helsinki.fi/10.1007/11940128_34.

DHLS06 Duchon, P., Hanusse, N., Lebhar, E. ja Schabanel, N., Could any graph be turned into a small-world? *Theoretical Computer Science*, 355,1(2006), sivut 96 – 103. URL http://www.sciencedirect.com/science/article/pii/S0304397505009187.

Kle00 Kleinberg, J., The small-world phenomenon: An algorithmic perspective. *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, New York, NY, USA, 2000, ACM, sivut 163–170, URL http://doi.acm.org.libproxy.helsinki.fi/10.1145/335305.335325.