

Implementacija Luvenovog algoritma upotrebom agentskih tehnologija

Uvod

Luvenov algoritam je metod za detekciju zajednica u grafovima. Grupu čvorova možemo da tretiramo kao zajednicu ako su ti čvorovi više povezani međusobno nego sa ostatkom mreže. Mera povezanosti (jačina zajednice) definisana je preko sledeće izvedene formule:

$$Q_c = \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2$$

Σ_{in} - sum svih grana između čvorova zajednice c

Σ_{tot} - sum svih grana čvorova zajednice c (uključujući i grane koje vode do čvorova u drugim zajednicama)

m - suma svih grana u celom grafu

c - zajednica

Ova formula naziva se modularnost zajednice, dok se modularnost čitavog grafa računa kao suma modularnosti svake zajednice:

$$Q = \sum_c Q_c$$

Luvenov algoritam nastoji da odredi zajednice, tako da maksimizuje ukupnu modularnost grafa.

Koraci algoritma

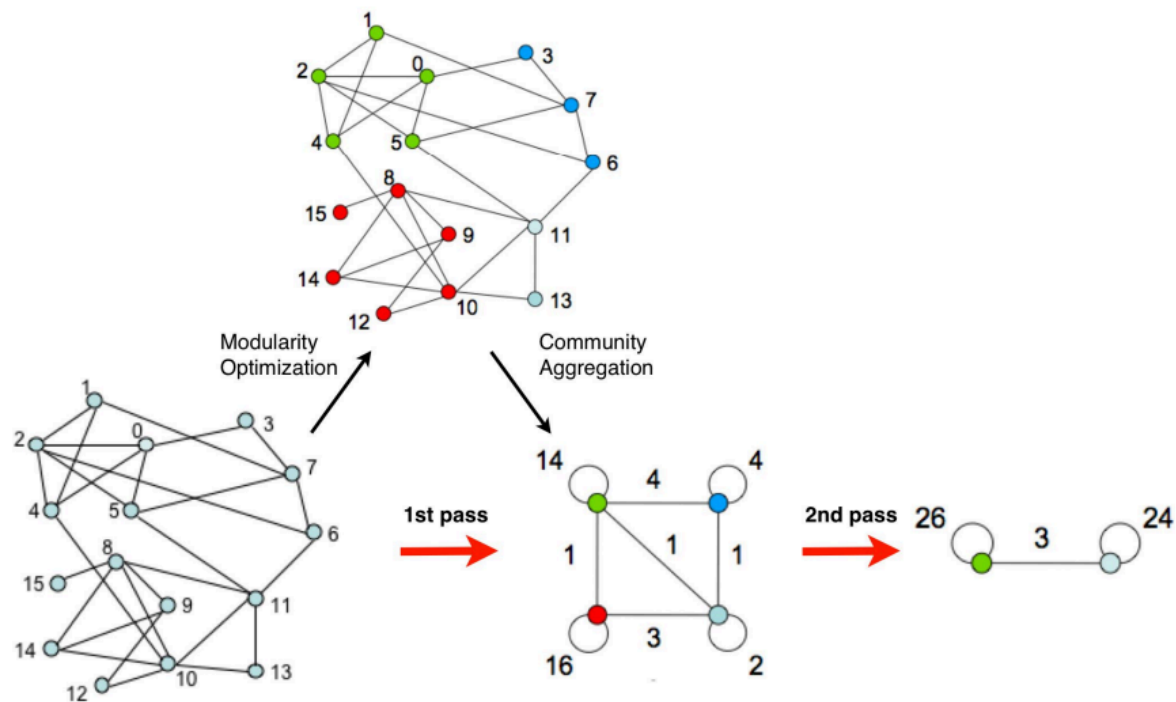
Algoritam se izvršava u dve faze koje se ponavljaju dok modularnost ne može više da se poboljša.

Faza 1: Lokalna optimizacija

1. Svakom čvoru se dodeljuje svoja zajednica na početku (svaki čvor je sam za sebe).
2. Za svaki čvor i , algoritam razmatra da ga premestiti u zajednicu jednog od svojih suseda.
3. Premeštanje se prihvata samo ako se modularnost poveća. Uzima se premeštanje čvora koje rezultuje maksimalnu modularnost.
4. Proces se ponavlja dok nije moguće dalje povećanje modularnosti pomeranjem pojedinačnih čvorova.

Faza 2: Agregacija zajednica

1. Sve zajednice iz prve faze se tretiraju kao pojedinačni "super-čvorovi".
2. Te "super-čvorove" povezuju veze između zajednica, a težine veza su zbir veza između čvorova iz različitih zajednica.
3. Nakon toga se ponovo primenjuje Faza 1 na novom, manjem grafu.



Vizuelni prikaz algoritma

Implementacija aktorskog sistema

Aktori i komunikacija između njih bi bili implementirani u Go programskom jeziku upotrebnom ručno pravljenog sistema za aktore. Takođe, želeli bismo da podržimo mogućnost da se ceo sistem izvršava na više mašina uz implementaciju cluster providera. Od dodatnih tehnologija planiramo da koristimo Docker i Docker Compose alat.

Tipovi aktora

- **Koordinatoriski aktor** - upravlja svim ostalima aktorima, prima poruke i vrši progresovanje sistema u naredene faze obrade
- **Particioni aktor** - upravlja grupom čvorova grafa, računa promenu modularnosti zajednice na osnovu prelaska čvora iz jedne zajednice u drugu
- **Agregatorski aktor** - prikuplja poruke od particionih aktora o novim granama, vrši sažimanje grana i kreiranje "super-čvorova"

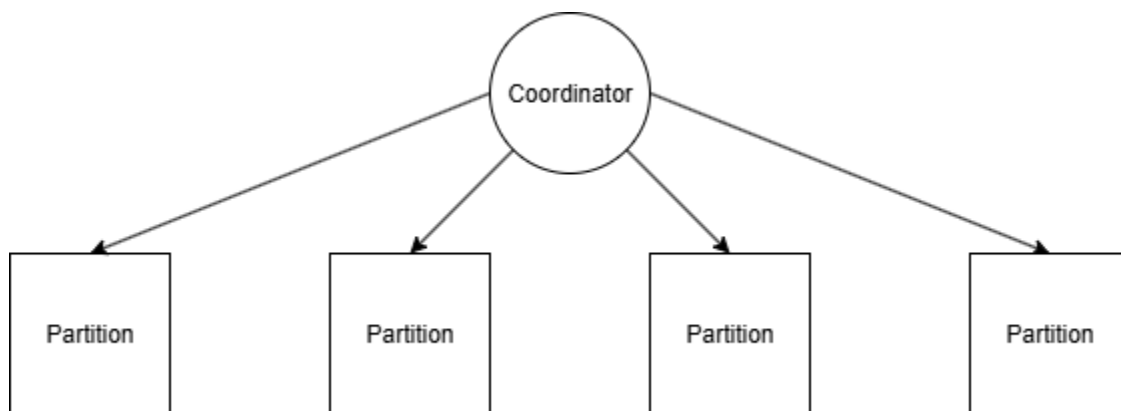
Koraci izvršavanja sistema

Faza 0: Inicijalizacija sistema

1. Kreiranje jednog koordinatora
2. Kreiranje K agregatorskih aktora
3. Kreiranje N particionih aktora
4. Particionisanje grafa i slanje podataka particionim aktorima

Faza 1: Lokalna optimizacija

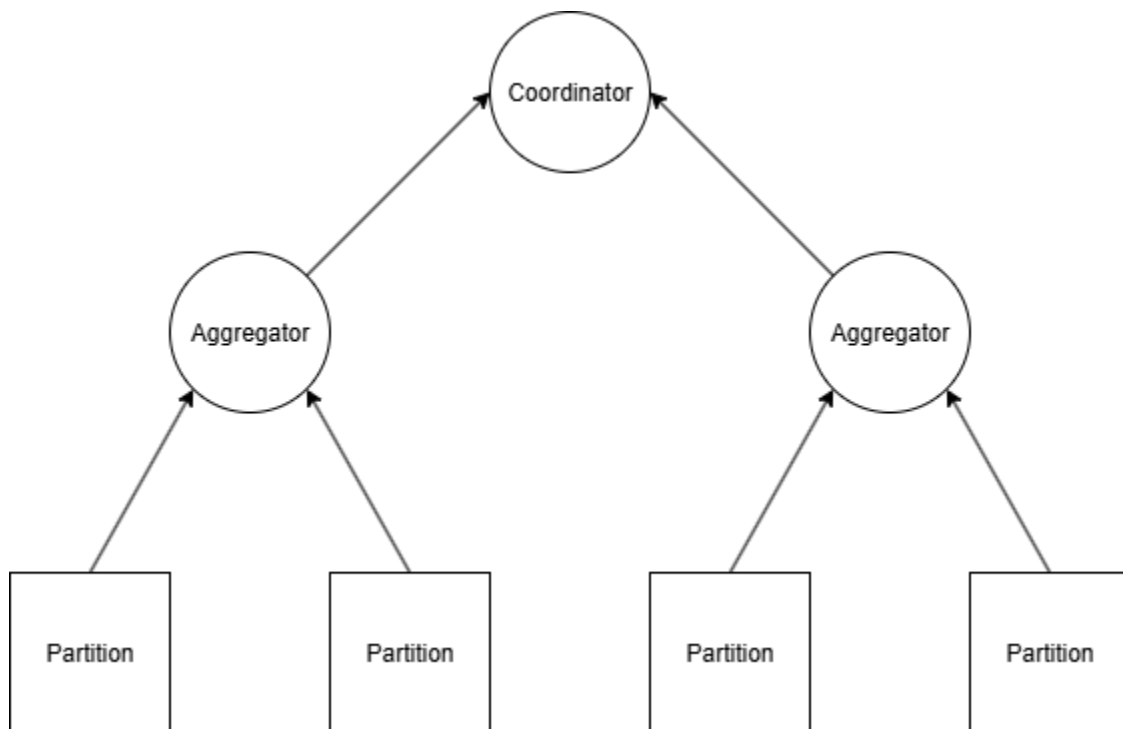
1. Koordinator saopštava da je počela faza 1 svim particionim aktorima.
2. Particioni aktor računa promenu modularnosti za promenu zajednice određenog čvora, komunicira sa ostalim particionim aktorima.
3. Koordinator na osnovu poruka od particionih aktora vodi globanu evidenciju o tome koji čvor je u kojoj zajednici i kolika je ukupna modularnost
4. Ukoliko nema promene ukupne modularnosti - algoritam se završava.



Faza 1: Lokalna optimizacija

Faza 2: Agregacija zajednica

1. Koordinator javlja svim particionim aktorima da je faza 1 završena.
2. Koordinator javlja svim particionim aktorima da je počela faza 2 - agregacija.
3. Particioni aktori vrše lokalnu agregaciju i uklanjanje grana. Koriste CRDT skup koji sadrži informacije koja zajednica je dodeljena kom čvoru. Redukovane podatke prosleđuju agregatorskim aktorima koji dalje vrše agregaciju i na osnovu zajednica kreiraju "super-čvorove".
4. Završetkom agregacije, koordinator saopštava redistribuciju novih podataka. Agregatori distribuiraju podatke između particionih aktera.
5. Ponavljanje faze 1. Ukoliko je graf znatno redukovano, neki particioni aktori mogu biti uništeni.



Faza 2: Agregacija zajednica

Tipovi poruka između aktora

- Koordinator → broadcast : početak faze 1
- Particioni akt. → Particioni akt. : zahtevanje stepena nekog čvora
- Particioni akt. → Koordinator i ostali particioni akt. : prelaz čvora i jedne zajednice u drugu (delta modularnost)
- Koordinator → broadcast : redukcija (lokalnai na nivou agregatora)
- Particioni akt. → Agregator : podaci o grani između “super-čvorova”
- Agregator → Koordinator : završena redukcija na nivou agregatora
- Koordinator → Svi agregatori : redistribucija
- Agregator → Particioni akt : nova grana redukovanoog grafa

Strukture podataka

- *Struktura za agregaciju težina grana (G-Counter CRDT)* - ova struktura se pravi za svaku granu i svaki element niza odgovara po jednom particionom akтору. Kada particioni aktor pošalje poruku za granu (u, v) sa težinom w, data težina se dodaje u polje niza. Ukupna težina između zajednica u i v će biti suma svih težina iz niza. Ova strukutra je replicirana po svakom agregacionom akтору.
- *Struktura za skupljanje prelaza čvorova (modifikovani Set CRDT)* - u ovu sktrukturu se dodaje uređena trojka (id_čvora, id_zajednice, delta_modularnost), Pri svakom dodavanju se zadržava samo onaj element koji ima veću delta_modularnost od trenutne. Ova strukutra je replicirana po svakom particionom akтору.
- Ostale načine čuvanja čvorova, grana, modularnosti i drugih među rezultata bismo razmotrili tokom same implementacije sistema.

Reference

- [Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. \(2008\). Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008\(10\), P10008](#)
- [Sattar, N. S., & Arifuzzaman, S. \(2018, August\). Parallelizing louvain algorithm: Distributed memory challenges. In 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress \(DASC/PiCom/DataCom/CyberSciTech\) \(pp. 695-701\). IEEE.](#)
- https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html

Autor

Nikola Kušlaković RA 8/2021