Grafu izmantošana datu reprezentēšanā un analīzē

Grafu paraugu ņemšana no statiskiem līdz laikā mainīgiem grafiem

Mārtiņš Dudelis

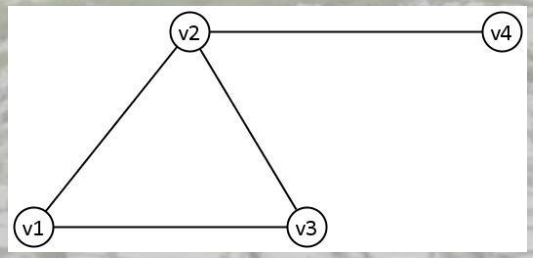
09.03.2016

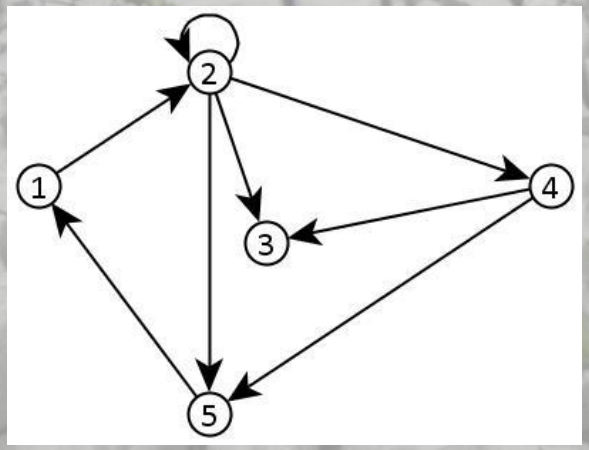
Grafs

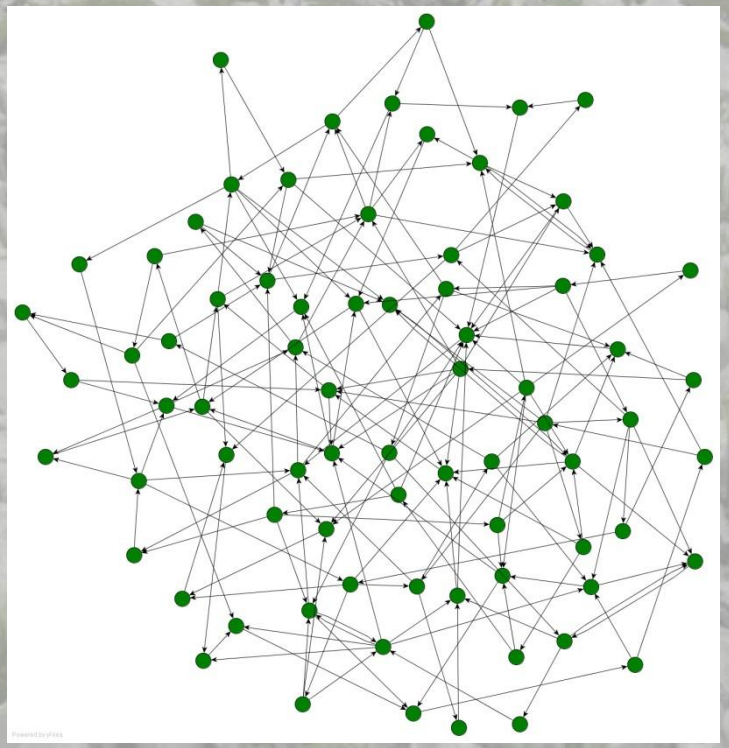
- **Neorientēts grafs** jeb saīsināti **grafs** ir kopu pāris **G = (V, E)**, kas apmierina $E \subseteq [V]^2$, t.i., E elementi ir kopas V divu elementu apakškopas.

- **Orientēts grafs** vai vienkārši **orgrafs** D sastāv no galīgas kopas V, kuras elementus sauc par virsotnēm, un galīgas kopas A no sakārtotiem atšķirīgu virsotņu pārīšiem, kurus sauc par lokiem. V(D) apzīmēsim par virsotņu kopu, bet A(D) par – loku kopu.

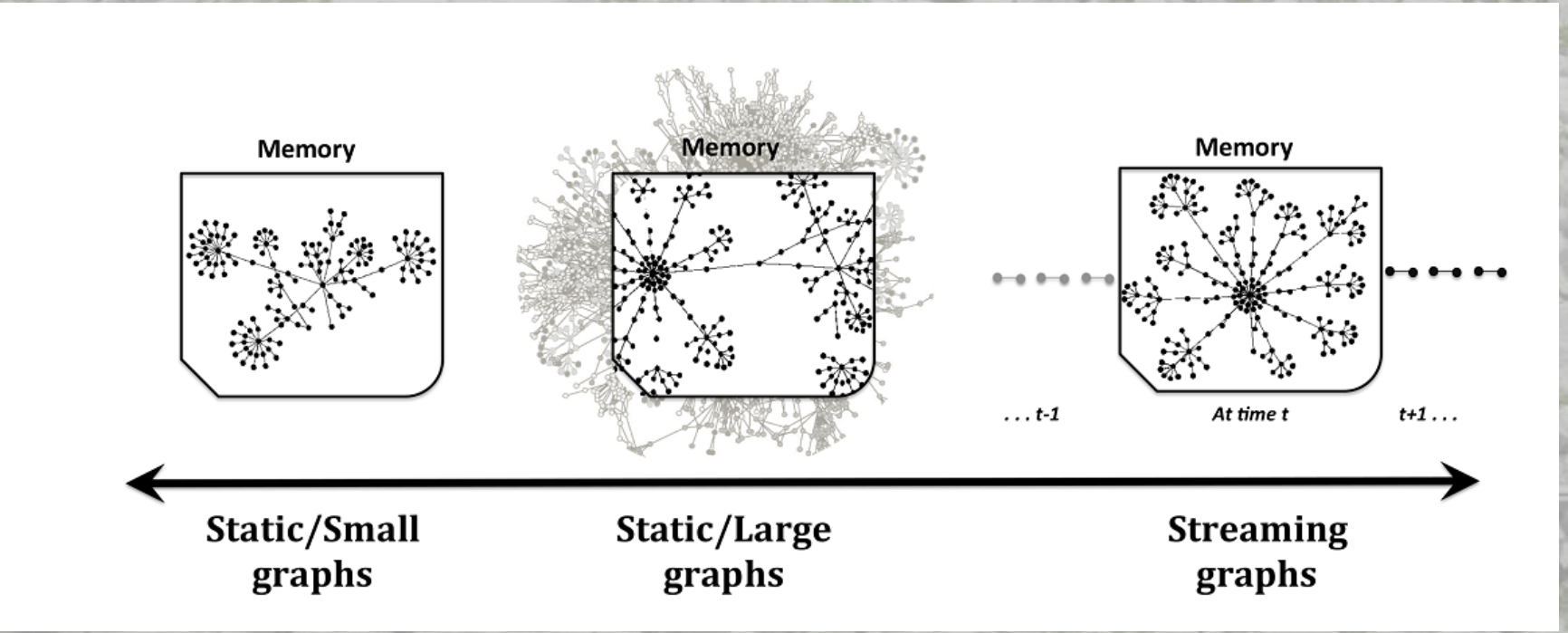
Grafu piemēri







No statiskiem grafiem līdz laikā mainīgiem grafiem



Grafu paraugu ņemšana

- Ar σ apzīmēsim jebkādu paraugu ņemšanas algoritmu, kas izmantojot gadījuma raksturu, izveido paraugu ***S*** no ***G (S= σ(G))***. Parauga kopa ***S*** var būt grafa ***G*** virsotņu apakškopa vai škautņu apakškopa, vai apakšgrafs.
- Parauga izmērs ***S*** tiek definēts kā relatīva grafa ***G*** daļa un apzīmē ar ***ϕ(0 ≤ ϕ ≤ 1)***. Vairumā gadījumu ***|S|= ϕ·|V|***.

Grafa parauga kvalitāte

- $S \subset V$ vai $S \subset E$. Tad, ja grafam G ir īpašība η un $\eta(S) \approx \eta(G)$, tad S tiek uzskatīts par labu grafa paraugu
- $S = G_S$ ir apakšgrafs $G_S=(V_S,E_S)$ no G. Tad grafa G topoloģisko īpašību kopa η_A un izpildās $\eta_A(S) \approx \eta_A(G)$, tad S tiek uzskatīts par labu grafa paraugu

Sasniedzamie mērķi

1. Novērtēt tīkla parametrus – virsotņu pakāpju sadalījums, ceļa garuma sadalījums, pudurēšanas koeficienta sadalījums
2. Iegūt reprezentējošu apakšgrafu
3. Novērtēt virsoņu atribūtus
4. Novērtēt škautņu atribūtus

1. un 2. fokuss uz visa tīkla raksturlielumiem

3. un 4. fokuss uz virsotņu vai škautņu īpašībām

Grafa paraugu ņemšanas algorimu tipi

- Grafu paraugu algoritmiem ir divi pamata soļi
  (1) Virsotņu ņemšana: $S = V_S$ no $G\ (V_s \subset V)$
  (2) Šķautņu ņemšana: $S = E_S$ no $G\ (E_s \subset E)$

Kad mērķis ir paņemt vai nu virsotnes vai škautnes (3.,4. vai 1.), tad izmanto vai nu 1 vai 2 soli.

Grafa paraugu ņemšanas algorimu tipi

- Ja mērķis ir iegūt apakšgrafu G_s, izmanto abus soļus. Šajā gadījumā škautņu izvēle ir ierobežota ar izvēlēto virsotņu kopu, lai veidotu ***inducēto apkšgrafu***.
- Izķir divu veidu ***inducētos apakšgrafus:***
  - Pilnīgi inducēts apakšgrafs
  - Daļēji inducēts apkašgrafs

Grafa paraugu ņemšanas algorimu tipi

- Virsotņu parauga ņemšana***(NS)*** - nesaglabā virsotņu pakāpju sadalījumu, grafa orģināl̄o sakarīgumu
- Škautņu parauga ņemšana***(ES)*** - iegūst daļēji inducētu apakšgrafu, nepietiekami labi saglabā vēlamas grafa īpašības – pudurēšanas koeficientu, grafa sakarīgumu. Saglabā labi īsāko ceļu sadalījumu.
- Uz topoloģiju bāzētas grafa paraugu ņemšanas algoritmi
  - Meklēšana plašumā ***(BFS)***
  - Gadījuma ceļi***(random walks)***
  - ***FFS***(forest fire sampling) – daļējs ***BFS***

Laikā mainīgi grafi

- ***Grafa plūsma*** ir sakārtota rinda ar škautnēm $e_{\pi(1)}$, $e_{\pi(1)}$, ..., $e_{\pi(M)}$, kur π ir jebkāda patvaļīga permutācija uz sķautņu indeksiem $M=\{1,2,...,M\}$, $\pi:[M]\rightarrow[M]$.
- Pamatā laikā mainīgi grafi(grafu plūsma) no statiskiem grafiem atškiras ar šadām trīs lietām:
  - Ļoti lielais šķautņu skaits laikā nav saglabājams atmiņā
  - Grafam var piekļūt tikai secīgi vienā piegājienā
  - Efektīva, reāllaika apstrāde ir kritiska

Laikā mainīgi grafi

Grafa plūsmas modelī, kad katra škautne $\boldsymbol{e} \in \boldsymbol{E}$ ierodas, paraugu ņemšanas algoritmam σ ir jāizlemj vai šo škautni iekļau vai nē. Paraugu ņemšanas algoritms σ var uzturēt stāvokli Ψ un izmantot to, lai izlemtu vai pievienot $\boldsymbol{e}$ paraugam vai nē.

Grafu plūsmas paraugu algoritma sareģītību mērā pēc:

1. Cik reizes tiek pārskatīta plūsma ω
2. Atmiņa, kas nepieciešama stāvoklim Ψ un izvadam
3. Cik raksturojošs ir izvadītais grafa plūsmas paraugs $\boldsymbol{S}$

Laikā mainīgi grafi

Paraugu ņemšanas algoritms no grafu plūsmas ir jebkurš grafu paraugu ņemšanas algoritms σ, kas izveidot parauga grafu G_s no atlasītajām škautnēm no ieejas grafa G secīgā kārtībā, vēlams vienā piegājienā $\omega = 1$, kamēr stāvoklis Ψ ir $\Psi \leq O(|G_s|)$.

Paveiktais

- Ir iegūti vairāki lielu tīklu dati pētījumu veikšanai
- Ir daļēji izveidota automatizēta ielādes un apstrādes sistēma šiem tīkla datiem
- Ir veikti sākotnējie eksperimenti ar statiskiem grafa paraugu algoritmiem

Tālākie darbi

- Turpināt datu ielādes un apstrādes, grafu algoritmu testēšanas sistēmas izstrādi
- Apkopot un uzrakstīt rakstu par statisku grafu paraugu ņemšanas algoritmiem un iespējamiem to uzlabojumiem
- Veikt apjomīgu grafa plūsmas paraugu veidošanas algoritmu testēšanu, apkopot rezultātus
- Pārbaudīt kādu iespaidu atstāj dažādi grafa plūsmas paraugu veidošanas algoritmi uz grafu algoritmu rezultātiem, ja šie algoritmi tiek darbināti uz iegūtajiem paraugiem.
- Apkopot un uzrakstīt rakstu par grafa plūsmas paraugu ņemšanas algoritmiem un iespējamiem to uzlabojumiem

Informācijas avoti

- N. Ahmed, J. Neville, and R. Kompella. Network Sampling: From Static to Streaming Graphs. ACM Transactions on Knowledge Discovery from Data, 2013.
- J. Pfeiffer III, J. Neville, and P. Bennett. Combining Active Sampling with Parameter Estimation and Prediction in Single Networks. In Proceedings of the Structured Learning: Inferring Graphs from Structured and Unstructured Inputs Workshop, ICML, 2013
- N. Ahmed, J. Neville, R. Rossi, and N. Duffield. Efficient Graphlet Counting for Large Networks. In Proceedings of the 15th IEEE International Conference on Data Mining, 2015.
- J. Pfeiffer III, J. Neville, and P. Bennett. Overcoming Relational Learning Biases to Accurately Predict Preferences in Large Scale Networks. In Proceedings of the 24th International World Wide Web Conference (WWW), 2015
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2010. 744 p.


Paldies par uzmanību!