Алгоритми за детекција на заедници врз граф заснован на Твитер граф со COVID-19 твитови за вистински и лажни вести

Мила Куч, 171074

Факултет за информатички науки и компјутерско инженерство, Скопје
Проектна задача по предметот Виртуелни Општества

Февруари, 2021 година

Абстракт - За полесна и поефикасна анализа на некои видови на податоци, особено тие кои потекнуваат од социјалните мрежи, графовите се совршената претстава која ќе ни помогне во согледување на многу врски и целокупната слика. Моделиран врз основа на податочното множество CoAID со COVID-19 твитови, овој проект ги согледува различните начини на формирање на т.н. заедници (communities) во склоп на самиот граф, нивната ефикасност и визуелизација.

I. Вовед

Еден од најсовршените примери за претставување на графови со податоци од вистинскиот свет се социјалните мрежи, каде јазел претставува корисник или профил, а раб претставува некаква врска помеѓу два корисника, како на пример понуда за пријателство, заследување, или интеракција на некаков пост. Но, во самите графови познат феномен е и заедницата.

Квалитативно, заедницата е дефинирана како подмножество од јазли во графот каде врските меѓу јазлите се погусти отолку врските со остатокот од мрежата. Детекцијата на заедниците генерално е наменета за мапирање на мрежата во дрво, каде листовите се јазлите, а гранките од повисоко ниво се групи од јазли, и така се идентификува хиерархиска структура. [1]

Во ера каде сите треба да бидеме информирани, вистинските информации се вреднуваат значително. Интересно е да се погледне како изгледа еден граф на социјален медиум од перспектива на вистински наспроти лажни информации, негова визуелна претстава, и да се погледне како се однесуваат алгоритмите за заедници во ваква средина како и да се донесат заклучоци за перформансите на истите. Во оваа проектна задача ќе се опфати токму тоа - да се разгледа таа зависност и да се евалуираат различните алгоритми.

II. Слични истражувања

Пристапот за анализа на еден граф може значително да варира во однос на тоа каква задача сакаме да извршиме над него, и која ни е целта за истиот.
Во случај на градење на заедници, постојат повеќе често користени алгоритми: Minimum-cut методот, хиерархиско кластерирање, Girvan–Newman алгоритмот, максимизација на модуларноста (Louvain методот), итн. [2]

Последната година имаше доста интересни пристапи во ова поле, како на пример решавање на овој проблем со граф невронски мрежи (Graph Neural Networks - GNNs) [3]. Друг пример вклучува експериментален пристап - Artificial Benchmark for Community Detection (ABCD) заснован на веќе постоечки алгоритам (LFR), за да се добијат подобри перформанси на повеќе податоци. [4]

Во врска со препознавањето на лажните вести, најчесто графовите се вклучуваат со граф конволуциските мрежи, комбинирајќи пристапи со обработката на природните јазици, како во истражувањето од страна на неколку факултети од Пакистан, Катар, и IBM Research - Almaden. [5]

III. Опис на проблемот

Дали има некаква поврзаност помеѓу создавањето на заедниците во граф и самата информација која ја носи јазелот, и како изгледаат самите заедници при различни алгоритми? Ова истражување е поттикнато од тоа прашање; како што беше претходно споменато, овој проект е базиран на актуелна тема - самите лажни наспроти вистински информации на социјалните медиуми.

Податочното множество, кое понатаму ќе биде елаборирано, е од социјалната мрежа Твитер, и врз основа на тоа множество ќе се изгради или симулира граф. Јазлите се всушност профилите, а врската меѓу нив е

Tweet-Retweet, односно споделување на постот со реална или лажна информација.

Неколку клучни точки во оваа проектна задача ќе бидат:

1. Да се изгради (симулира) граф со достапните податоци од податочното множество
2. Да се визуелизира графот - кои јазли имаат вистински вести, а кои лажни
3. Да се испитаат како сите алгоритми за детекција на заедници се однесуваат во склоп на овој проблем, и соодветно да се визуелизираат
4. За секој алгоритам да се согледа дали има некаква врска помеѓу типот на јазел(вистински и лажен) и тоа како се формираат заедниците.

IV. Податочно множество

Податочното множество CoAID [6] се состои од 5.216 новости, 296.752 активности на корисници, 958 објави на социјалната платформа Твитер за COVID-19. Самото податочно множество е поделено на неколку дела, т.е. податоци кои се собрани секој втор месец (мај, јули, септември и ноември). Податочното множество е достапно на Github [7] и од таму може да се преземе.
Со помош на Hydrator ги преземав податоците од секој твит, според неговото ID, а потоа од JSON формат ги претворив во CSV фајлови.

Првата замисла за проектот беше со помош на Twitter API да се најдат id броевите на сите ретвитови, и истите да бидат реални податоци кои ќе го сочинуваат графот. Но, со масовните нови рестрикции на Твитер, над 170 000 профили оваа година беа избришани [8], што значи дека голем дел од ретвитовите беа невозможни да се најдат. (Сл. 1)

```
18      try:
19          retweets_list = api.retweets(numb)
20          for retweet in retweets_list:
21              print(retweet.user.display_name)
22      except:
23          print("That tweet does not exist!")
24      count+=1
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
That tweet does not exist!
```

Сл. 1 Дел од кодот за преземање ретвитови од твит

Алтернативно решение е да се направи симулација на графот т.е. да се земат информации за секој јазел колку retweets има, и соодветно да се изгради граф.
Ова решение го постигнав со едноставен код за генерирање на јазли и рабови:

1. Се читаат ids и бројот на ретвитови од податочното множество (колоните ‘id’ и ‘retweet_count’)
2. Се отстрануваат аутлаерите т.е. вредности кои се премногу надвор од нормалата и би пречеле во нашата задача - на пример: 22461 или 34768, и нивната вредност се заменува со некоја произволна горна граница, која може ние да ја поставиме.
3. За секој јазел се генерира едноставно id (како замена за неговото
4. За секој јазел со кој е поврзан се генерира id во вредност од 0 до некој коефициент. Коефициентот ние го задаваме и може да се менува, со што ако е помал ќе имаме поврзан граф, а ако е поголем графот има повеќе одвоени компоненти.
5. Чекор 3 и чекор 4 всушност ги формираат рабовите во графот
6. Рабовите од чекор 3 се запишуваат во документ, за да можат да се вчитуваат за обработката на проблемите. Ова се повторува и за вистинските и за лажните твитови.

За полесна визуелизација и манипулација со податоците, се земаат првите 20 твитови од податочното множество. Со крајот на овој код се добиваат два документи - *small_scale_fake*.txt и *small_scale_real*.txt, со рабовите од двата графови, со лажни и реални информации соодветно. За референца документите со кои ќе работам на проектот се поставени подолу. [9][10]

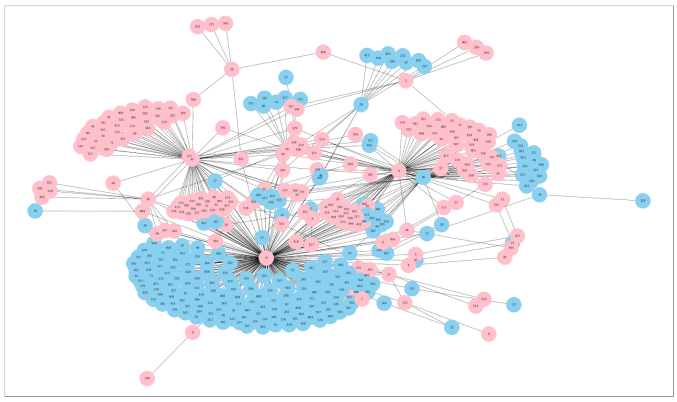
V. Подготвување на граф

За манипулација со податоците, се користат неколку библиотеки; Networkx како една од попопуларните и поуспешните во полето на работа со графови, како и Pandas и Numpy како стандардни библиотеки за работа со податоци.

Пред сè, двата графови (за реални и лажни вести) се вчитуваат од соодветните документи; и на сите јазли од графовите се придружува лабелата ‘Fake’ или ‘Real’, соодветно, како податок што се чува како атрибут на самиот јазел. Потоа тие се спојуваат во еден граф - и се наоѓа најголемата сврзана компонента од нив, бидејќи со неа треба да работиме.

Како графот изгледа е прикажано на Сл.2, а истото се постигнува со функција за визуелизација, во која е искористен методот *draw_networkx* од библиотеката Networkx. Дополнително, јазлите со лажните вести (секој

оригинален лажен твит и неговиот ретвит) се обележани со розева боја, а соодветно, тие со реалните вести со сина Во графот се наоѓаат вкупно 399 јазли и 532 рабови.



Сл.2 Визуелизација на граф - лажни и вистински вести

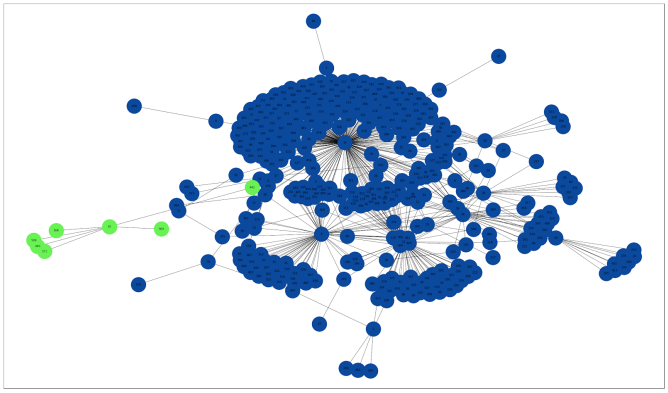
VI. Алгоритми за детекција на заедници

Пред да преминеме на алгоритмите, треба да се дефинираат две функции кои ќе бидат од корист во наредните чекори:

1. Функцијата *plot_subgraphs* - која од листа на заедници создава подграфови во даден граф, и ги исцртува секој со различна боја генерирана во HEX формат. И оваа функција го користи методот *draw_networkx* од библиотеката Networkx.
2. Функцијата *evaluation* која на влез прима листа на заедници, и соодветно проверува колку проценти од јазлите во секоја заедница припаѓаат на 'Real', а колку на 'Fake'. Оваа функција ќе ни помогне да согледаме дали постои зависност меѓу начинот на групирање на јазлите во заедници и нивната лабела при секој алгоритам.

А. Girvan-Newman алгоритам

Girvan–Newman алгоритмот (наречен по Michelle Girvan и Mark Newman) е хиерархиски метод кој се користи за детекција на заедници во комплексни системи. Овој алгоритам детектира заедници со тоа што прогресивно ги отстранува рабовите од оригиналната мрежа. Сврзаните компоненти од преостанатата мрежа се заедниците; наместо да се обиде да конструира мерка која посочува кои рабови се најцентрални во заедниците, овој алгоритам се фокусира повеќе на рабовите кои се "измеѓу" заедници.[11]
Овој алгоритам во проектот се извршува преку методот *girvan_newman* од *networkx.algorithms.community*.[12] Во нашиот граф овој алгоритам пронајде 2 заедници, првата од 385 јазли, втората од само 7. Графот може да се види на Сл.3



Сл.3 Girvan-Newman алгоритам за детекција на заедници

При повикување на функцијата за евалуација на овој алгоритам, интересно е да се забележи дека заедницата со поголемиот број на јазли е од мешан вид, а додека пак помалата заедница е чисто од лажни вести. (Сл.4)
За овој алгоритам бројот на заедници е премал за да се направи заклучок за тоа дали видот на јазлите зависи за тоа како се поделени.

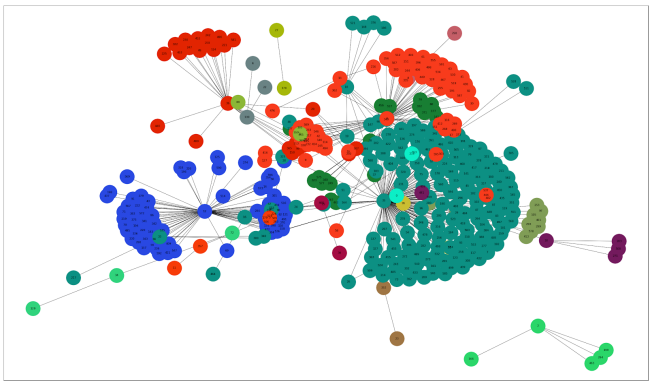
```
Fake:46.493506493506494%  Real:53.5064935064935%
Fake:100.0%  Real:0.0%
```

Сл.4 Girvan-Newman - Резултати од евалуација на видот на јазлите во заедниците

Б. Label propagation заедници

Label Propagation алгоритмот кој на некој начин имитира епидемиска контаминација со тоа што се прошируваат лабелите, е популарен алгоритам за детекција на заедници. Неговите варијанти (надоградувања, надополнувања, комбинации) се стремат да го подобрат или да го адаптираат, додека ги зачувуваат предностите на оригиналната формулација.[13] Дополнително, не може да се имплементира на насочени графови.

Овој алгоритам во проектот се извршува преку методот *label_propagation_communities* од *networkx.algorithms.community.label_propagation*. [14]
Резултатот е сочинет од дури 18 заедници, и може да се види на Сл.5.



Сл.5 Label Propagation алгоритам за детекција на заедници

Евалуацијата, пак, дава интересни резултати. Во овој случај, ретки се заедниците кои не содржат мнозинство од еден вид јазли. Такви има само 2 на број, поделени по 50% на вистински и лажни, а сите останати заедници се или целосно од еден вид, или имаат значително мнозинство. Резултатот може да се види на Сл. 6.

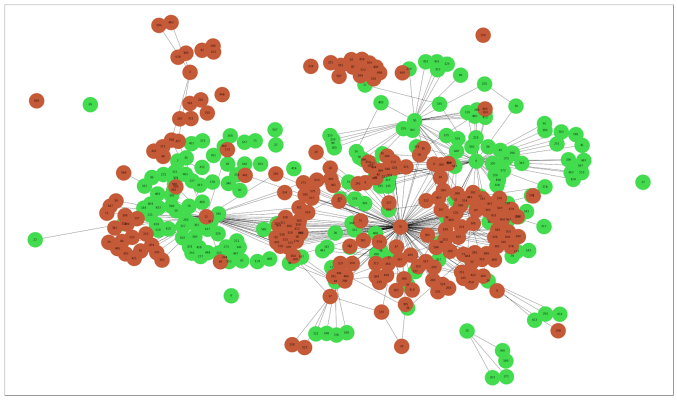
```
Fake:100.0%  Real:0.0%
Fake:0.0%  Real:100.0%
Fake:0.0%  Real:100.0%
Fake:11.11111111111111%  Real:88.88888888888889%
Fake:66.66666666666666%  Real:33.33333333333333%
Fake:100.0%  Real:0.0%
Fake:0.0%  Real:100.0%
Fake:14.117647058823529%  Real:85.88235294117646%
Fake:4.0%  Real:96.0%
Fake:100.0%  Real:0.0%
Fake:100.0%  Real:0.0%
Fake:0.0%  Real:100.0%
Fake:50.0%  Real:50.0%
Fake:50.0%  Real:50.0%
Fake:100.0%  Real:0.0%
Fake:100.0%  Real:0.0%
Fake:100.0%  Real:0.0%
Fake:100.0%  Real:0.0%
```

Сл.6 Label Propagation - Резултати од евалуација на видот на јазлите во заедниците

B. Kernighan-Lin Bisection

Kernighan–Lin алгоритмот е хевристички алгоритам за наоѓање на партиции на граф, а има важна улога при дигиталните кола и компоненти на VLSI.[15] Алгоритамот работи на тој начин каде се дели графот на 2 дела - А и Б, кои се делат на партиции, а деловите А и Б понатаму можат да си ги менуваат тие партиции додека не се дојде до најоптималното решение. [16]

Овој алгоритам во проектот се извршува преку методот *kernighan_lin_bisection* од *networkx.algorithms.community.kernighan_lin*. [17] Како резултат се добиваат 2 поголеми заедници, од точно ист број на јазли - 196 во секоја, а визуелизацијата може да се види на Сл.7.



Сл.7 Kernighan–Lin алгоритам за детекција на заедници

Во врска со евалуацијата, може од резултатот да се заклучи дека на овој алгоритам не влијае лабелата, бидејќи двете заедници имаат приближно ист процент на јазли од двата типа, како што може да се види на Сл.8

```
Fake:46.42857142857143%  Real:53.57142857142857%
Fake:48.46938775510204%  Real:51.53061224489795%
```

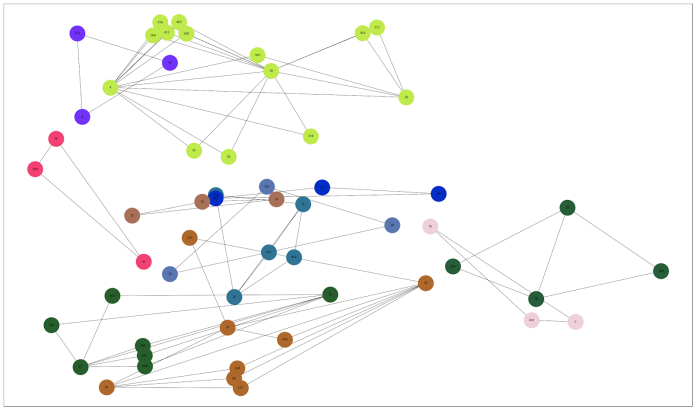
Сл.8 Kernighan–Lin - Резултати од евалуација на видот на јазлите во заедниците

Г. k-clique заедници

Овој метод ги гради заедниците од k-cliques (“дружини”), кои се однесуваат на целосни (целосно поврзани) подграфови од k јазли. (На пример k-clique со k = 3 е еквивалентно со триаголник). Заедницата е дефинирана како максимална унија од k-clique елементи кои можат да се достигнат еден од друг низ серија на соседни k-cliques(тие кои делат k − 1 јазли). [18]

Овој алгоритам во проектот се извршува преку методот *k_clique_communities* од *networkx.algorithms.community*. [19]

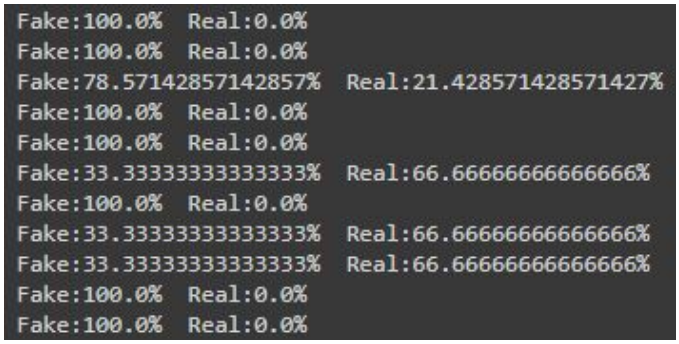
Резултатот се 22 подграфови, а некои од нив се празни и затоа не се земаат во предвид ни во визуелизацијата ни во евалуацијата. Така, остануваат 11 компоненти со реални елементи. Тие можат да се видат на Сл.9



Сл.9 K-Clique алгоритам за детекција на заедници

Конечно, евалуацијата на овој последен алгоритам покажува дека во повеќето јазли доминира еден тип на лабела со поголем процент, слично како и кај Label Propagation алгоритмот.

Резултатите може да се видат на Сл.10



Сл.10 K-Clique- Резултати од евалуација на видот на јазлите во заедниците

VII. Заклучок

Од сите резултати може да се забележи дека кај Label Propagation и K-Clique видот на јазлите се зема во предвид кога се делат на заедници, додека пак во Girvan–Newman и Kernighan–Lin тоа не е случајот или не може да се одреди. Сепак, ова е само една симулација на граф и резултатите можат да варираат доколку истите алгоритми се извршат на друго податочно множество, друг граф или во друга ситуација.

Целокупниот процес на анализа и испитување има доста место за подобрување, како и процесот на генерирање на графот од податоците. Кодот на проектот е поставен на Google Colab заедно со сите резултати и визуелизации.[20]

Референци:

[1]https://www.pnas.org/content/101/9/2658#:~:text=Qualitatively%2C%20a%20community%20is%20defined,the%20rest%20of%20the%20network.&text=Starting%20from%20the%20set%20of,in%20order%20of%20decreasing%20weight

[2]https://en.wikipedia.org/wiki/Community_structure

[3]https://arxiv.org/abs/1705.08415

[4]https://arxiv.org/abs/2002.00843

[5]https://www.researchgate.net/publication/346571571_Fake_News_Detection_in_Social_Media_using_Graph_Neural_Networks_and_NLP_Techniques_A_COVID-19_Use-case

[6]CoAID: COVID-19 Healthcare Misinformation Dataset, Limeng Cui and Dongwon Lee, 2020, 2006.00885, arXiv, cs.SI

[7]https://github.com/cuilimeng/CoAID

[8]https://edition.cnn.com/2020/06/11/tech/twitter-manipulation-account-removal/index.html

[9]https://drive.google.com/file/d/1-TdJXZ9eMQI6zi0hueoL_oe0cmVeRnu0/view?usp=sharing

[10]https://drive.google.com/file/d/1-L87vK9_7PttmOlBxaV6G6ReQMBFV_yU/view?usp=sharing

[11]Community structure in social and biological networks
M. Girvan, M. E. J. Newman
Proceedings of the National Academy of Sciences Jun 2002, 99 (12) 7821-7826; DOI: 10.1073/pnas.122653799
https://www.pnas.org/content/99/12/7821

[12]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.centrality.girvan_newman.html#networkx.algorithms.community.centrality.girvan_newman

[13]Sara E. Garza, Satu Elisa Schaeffer,
Community detection with the Label Propagation Algorithm: A survey,Physica A: Statistical Mechanics and its Applications,Volume 534,2019,122058,ISSN 0378-4371,
https://doi.org/10.1016/j.physa.2019.122058.

[14]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.label_propagation.label_propagation_communities.html#networkx.algorithms.community.label_propagation.label_propagation_communities

[15]https://en.wikipedia.org/wiki/Kernighan%E2%80%93Lin_algorithm

[16]https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/kernlin.pdf

[17]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.kernighan_lin.kernighan_lin_bisection.html#networkx.algorithms.community.kernighan_lin.kernighan_lin_bisection

[18]https://en.wikipedia.org/wiki/Clique_percolation_method

[19]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.kclique.k_clique_communities.html#networkx.algorithms.community.kclique.k_clique_communities

[20]https://colab.research.google.com/drive/1RnSMhUp_BURtubJsjLCQQV3XJ9jgv5aQ?usp=sharing