Carnegie Mellon University
15-826 Multimedia & Data Mining
Fall 2024 - C. Faloutsos

Project Description: `GraphSpot`

Abstract

Short version: for **Phase 1**, check subsection 4-Phase-1, i.e.: do literature survey and provide a few preliminary plots.

1 Introduction - Problem description

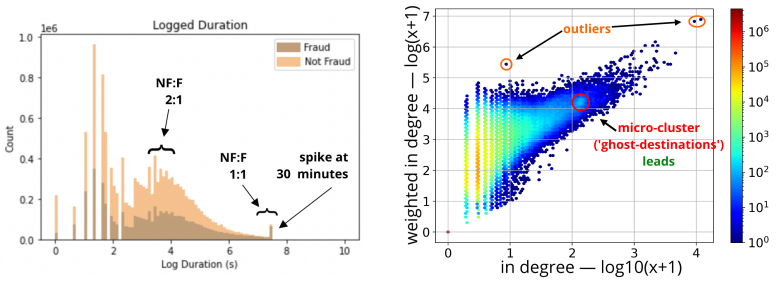
Given a graph (like who-calls-whom in a telephone network, or who-likes-whom in Facebook; or who-reviews-what in Yelp/TripAdvisor), can you find fraudsters, or, at least, suspicious entities?

We have the following goals in this project

1. **Spotting strange behaviors**: find suspicious entries, in some real graph datasets that we will provide.
2. **Tool development**: develop tools to make the above task easier for you, as well as the multiple colleagues in the industry, academia and government, that have to analyze such graphs. We envision a re-implementation/generalization of the *CallMine* system [4]. (see Figure 1). The goal is to *justify* our responses, and *visualization* provides strong, convincing arguments.

Motivation: Graphs appear in numerous settings; spotting anomalies and fraudsters is vital. Some settings include:

- *who-friends-whom* on FaceBook. Fraudster may 'buy' friends, from unscrupulous companies, so that they seem more important than they actually are. Similarly, fraudsters may buy 'likes'
- *who-follows-whom* on Twitter. Similarly, fraudsters 'buy' followers, to boost their importance, and the rate they charge for advertizers
- *who-reviews-what* on Yelp, ebay, amazon, tripAdvisor: dishonest sellers may 'buy' fake reviews
- *who-calls-whom*: telemarketers in phone networks, would probably have different behavior than normal users
- *fake-news*: fraudsters re-tweeting fake news, would probably form dense subgraphs (all retweeting each others tweets, so that they all look important)



(a) histogram (b) heat-map

Figure 1: Screenshot of the '*CallMine*' system [4] that we want to generalize in this project. (a) histogram of call durations. (b) heatmap of in-degree vs weighted in-degree. See the original paper for more details.

- *health-insurance* fraud: groups of fraudulent doctors, submit similar diagnoses (and expenses), for too many patients.
- *human trafficking* detection: escort-service advertisements, look too similar to each other, if they come from organized crime.

2 Data

All the datasets will be on 'box' in this URL. You will be 'invited' to access it after you sign an NDA (non disclosure agreement).

Phase 1: We will provide a small (2GB) real dataset, directed, weighted, containing phonecall information for a single day from a large anonymous telecom. It will be a CSV file with `(origin, destination)` pairs, and additional information for each phonecall (like timestamp, call duration, ringing duration etc). Some of these phonecalls are labeled as 'fraud'.

Phase 2 We will provide additional datasets, for phonecalls during subsequent days. These will be CSV files with the same schema.

3 Paper list for your survey

Please read

- **Part I** of the graph mining textbook, and
- the *CallMine* paper [4]

No need to comment on these two items.

In addition, please choose at least 3 papers from the list below, and comment on them.

3.1 Papers for your survey

Anomaly detection, and scoring

1. Graph anomaly detection survey by Leman Akoglu et al. [2]
2. OddBall paper (Akoglu et al) [1]. Spots nodes that have strange 'ego-nets', in weighted or unweighted settings.
3. isolation forests (Liu+, ICDM'08) [13]. Gives a 'weirdness' score to each point in a k-dimensional cloud of points.
4. random cut forests Guha+, ICML'16. [8] Similar to Isolation Forests.
5. Gen2out Lee+ IEEE Big Data'21 [12] Code here Spots micro-clusters (ie, group-anomalies), in addition to isolated anomalies.

Spectral methods for lock-step behavior

1. Spectral methods: EigenSpokes and the 'SpokEn' algorithm; [14] 'LockInfer' follow-up algorithm. Both spot groups of nodes that have similar behavior, which is usually suspicious.
2. ND-Sync - summary outside paywall (Giatsoglu+, PAKDD'15) [5] and also [6]. Algorithms to spot strange groups of twitter users.
3. f-Box [16]. Complements spectral methods, spotting small groups of suspicious nodes that may be missed otherwise.

Dense-block detection

1. Dense-block detection algorithms: Fraudar (Hooi+, KDD'16) [10] D-cube [18] and M-zoom (Kijung Shin et al) [17] Like the spectral methods, but have simpler algorithms, often have better accuracy, and give probabilistic performance guarantees.
2. CopyCatch (Beutel+, www13) [3]. Finds nodes with lock-step behavior, taking timestamps into account.
3. CatchSync (Jiang+, kdd'14) [11]. Finds groups of nodes that are (a) too similar to each other and (b) too different from everybody else.
4. Generalized Means [19] Novel measures for dense-block detection.

Applications, explainability, visualization

1. Human-trafficking detection (Rabbany+, KDD'18) [15].
2. Spectral Lens Goebl+, ICDM'17 [7] Focuses on weighted graphs.
3. LookOut (Gupta+, PKDD'18) [9]. Gives algorithms to visually justify the outliers that, say, isolation forests, have discovered.
4. Graph-theoretic Scagnostics [20] How to find the most interesting scatter-plots.

4 Tasks and Deliverables

Here is the detailed list of deliverables and point distribution. The maximum grade in each phase is 100, and the weights of each phase are as announced (10%, 10% , 80% of the project grade, or equivalently, 4%, 4% and 32% of the course grade).

Phase 1: 4% of course grade, max score: 100

Your write-up should be about 6-8 pages.

1. (**60 pts**) **Survey:** Complete a literature survey: at least *3 paper reviews*, from the introductory papers above (Section 3). Paper reviews should consist of the following:
   (a) the *problem definition* that the paper is addressing
   (b) a *summary* of the main idea of the paper (in your *own* words - cutting-and-pasting text from the paper or any other source, is **plagiarism**)
   (c) whether/why it is *useful* for your `GraphSpot` project.
   (d) list of *shortcomings*, that you think that future research could address.
2. (**40 pts**) **Preliminary suspects:** Among the plots that *CallMine* provides (in-degree distribution, out-degree distribution, heatmaps, etc), inspect them, and try to spot anomalies, that is, strange nodes that have not been labeled as 'fraud' (yet).
   Report the k=2 most suspicious nodes that you have found,
   - Give the list of such nodes, grouped, if they form natural groups (eg., nodes of a suspicious near-clique should be reported together; similarly, the nodes of a suspicious chain, etc).
   - Provide the plot(s) or other evidence, that supports your suspicion.

   FYI: Verification by expert: The instructor plans to ask the data owner, Dr. Pedro Fidalgo of Mobileum, so that he and his group can check the nodes that you provide, to verify whether they are indeed fraudsters or not.

Phase 2: 4% of course grade, max score: 100

Your write-up should be about 10-15 pages (**including** your Phase 1 write-up)

1. (**50 pts**) **More features:** Design and implement f=3 (or more) edge-based features: For each pair (origin,destination), try to characterize the behavior of 'origin' calling 'destination'. Possible such features could be: count of phonecalls, total/average/median/stdv duration of phonecalls, inter-arrival time (median,stdv), count of phonecalls per hour of day (9am, 10am, etc), uniformity/burstiness of activity (using some version of entropy), or anything else you may find promising.
2. (**50 pts**) **Larger datasets:** Run *CallMine* and your edge-feature code on the larger dataset (spanning several days, and several GB) and report anomalies:
   - give the list of top k=2 most suspicious node-ids or edge-ids
   - justify our decision, with words and with plots.

   Again, the instructor plans to send your lists to Dr. Fidalgo for verification.

Phase 3: 32% of course grade, max score: 100

Your write-up should be about 20-30 pages long (including all previous write-ups).

1. (**10pts**) **Rationale:** Justification for the earlier edge-features you used. (Eg., "I chose median inter-arrival time, to capture repetitive behavior of possible telemarketers")

2. (**10pts**) **Vindication:** Expert's feedback - If Dr. Fidalgo provides feedback on time for your responses of the list of suspects, mention his response ('fraud' or 'not')
3. (**50pts**) **Additional features:** Implementation of additional f'=4 more edge-base features; Justification of your choice of features.
4. (**30pts**) **Additional suspects:** Again, for your new f' edge features, give the top k=2 most suspicious edges, along with your evidence (plots, etc).

5 Details on deliverables and software packaging

5.1 Check-lists on deliverables:

For every phase, please:

1. Hand-in a hard copy of your write-up, typed, 12pt font, neat and with pictures if applicable

More details:

1. Use the LaTeX template at:
   `http://www.cs.cmu.edu/~christos/courses/826-resources/PROJECT-SAMPLES/samplePaper.tar.gz`
   Adapt the section headers, accordingly, eg.,
   - introduction
   - ph1: Survey
   - ph1: Preliminary suspects
   - ph2: More features
   - ph2: Larger datasets
   - ...
2. Check grammar and syntax (small penalty for each typo/grammar error).
3. Keep the graded reports and attach them, every time. That is for Phase 2, attach the graded Phase 1 report; for Phase 3, attach all previous, graded, reports.

5.2 Logistics - reminders

- *Academic Attribution / Plagiarism*: Whenever you use ideas, text, code, algorithms, from someone else, *please cite this person, paper, or url*. Copying without attribution constitutes **plagiarism** leading to severe penalties (failing the class, expulsion, etc).
- *LLMs/ChatGPT/etc*: As mentioned in the NDA that you have to sign, we are **not allowed** to use any part of the datasets as a prompt to a public LLM, which may keep a local copy to train on.

References

[1] L. Akoglu, M. McGlohon, and C. Faloutsos. oddball: Spotting anomalies in weighted graphs. In *PAKDD (2)*, volume 6119 of *Lecture Notes in Computer Science*, pages 410–421. Springer, 2010.

[2] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Discov.*, 29(3):626–688, 2015.

[3] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, pages 119–130. International World Wide Web Conferences Steering Committee / ACM, 2013.

[4] M. T. Cazzolato, S. Vijayakumar, M. Lee, C. Vajiac, N. Park, P. Fidalgo, A. J. M. Traina, and C. Faloutsos. Callmine: Fraud detection and visualization of million-scale call graphs. In *CIKM*, pages 4509–4515. ACM, 2023.

[5] M. Giatsoglou, D. Chatzakou, N. Shah, A. Beutel, C. Faloutsos, and A. Vakali. Nd-sync: Detecting synchronized fraud activities. In *PAKDD (2)*, volume 9078 of *Lecture Notes in Computer Science*, pages 201–214. Springer, 2015.

[6] M. Giatsoglou, D. Chatzakou, N. Shah, C. Faloutsos, and A. Vakali. Retweeting activity on twitter: Signs of deception. In *PAKDD (1)*, volume 9077 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2015.

[7] S. Goebl, S. Kumar, and C. Faloutsos. Spectral lens: Explainable diagnostics, tools and discoveries in directed, weighted graphs. In *ICDM*, pages 877–882. IEEE Computer Society, 2017.

[8] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust random cut forest based anomaly detection on streams. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2712–2721. JMLR.org, 2016.

[9] N. Gupta, D. Eswaran, N. Shah, L. Akoglu, and C. Faloutsos. Beyond outlier detection: Lookout for pictorial explanation. In *ECML/PKDD (1)*, volume 11051 of *Lecture Notes in Computer Science*, pages 122–138. Springer, 2018.

[10] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. FRAUDAR: bounding graph fraud in the face of camouflage. In *KDD*, pages 895–904. ACM, 2016.

[11] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Catchsync: catching synchronized behavior in large directed graphs. In *KDD*, pages 941–950. ACM, 2014.

[12] M. Lee, S. Shekhar, C. Faloutsos, T. N. Hutson, and L. D. Iasemidis. Gen^{2}out: Detecting and ranking generalized anomalies. In *IEEE BigData*, pages 801–811. IEEE, 2021.

[13] F. T. Liu, K. M. Ting, and Z. Zhou. Isolation forest. In *ICDM*, pages 413–422. IEEE Computer Society, 2008.

[14] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD (2)*, volume 6119 of *Lecture Notes in Computer Science*, pages 435–448. Springer, 2010.

[15] R. Rabbany, D. Bayani, and A. Dubrawski. Active search of connections for case building and combating human trafficking. In *KDD*, pages 2120–2129. ACM, 2018.

[16] N. Shah, A. Beutel, B. Gallagher, and C. Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In *ICDM*, pages 959–964. IEEE Computer Society, 2014.

[17] K. Shin, B. Hooi, and C. Faloutsos. M-zoom: Fast dense-block detection in tensors with quality guarantees. In *ECML/PKDD (1)*, volume 9851 of *Lecture Notes in Computer Science*, pages 264–280. Springer, 2016.

[18] K. Shin, B. Hooi, J. Kim, and C. Faloutsos. D-cube: Dense-block detection in terabyte-scale tensors. In *WSDM*, pages 681–689. ACM, 2017.

[19] N. Veldt, A. R. Benson, and J. M. Kleinberg. The generalized mean densest subgraph problem. In *KDD*, pages 1604–1614. ACM, 2021.

[20] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *INFOVIS*, pages 157–164. IEEE Computer Society, 2005.