Chapter 1

Fraud Detection - Dense Subgraph Detection

Contributed by Tong Zhao

1.1 Introduction

Fraud behaviors can be spotted everywhere on online applications such as social networks where the behavior data can be represented as large bipartite graphs. These graphs consist of links between followers and followees. Fraudulent actions such as fake followers usually result with creating large and dense subgraphs. For example, as a large number of follower buyers buy followers from one major follower seller, these follower buyers togethor with the bot followers controled by the seller will form a subgraph with high density. Hence many existing detection methods [15, 27, 25] estimate the suspiciousness of users by identifying whether they are within a dense subgraph.

1.2 The Problem as a Graph

Here we define the definitions of density for graphs according to [4, 12, 18].
Let $G = (V, E)$ be a undirected graph with vertices V and edges $E \subseteq V \times V$. $E(V)$ stands for the set of edges induced by V, that is

$$E(V) = \{(i, j) \in E : i \in V, j \in V\}$$

Then the density of subgraph induced by $S \subseteq V$ can be defined as

$$d(S) = \frac{|E(S)|}{|S|}$$

Note that $2d(S)$ is actually the average degreee of the subgraph induced by S. The Densest Subgraph problem can be defined as

$$DS(G) = \max_{S \subseteq V} \{d(S)\}$$

For directed graphs. Let $G = (V, E)$ be a directed graph with vertices V and edges $E \subseteq V \times V$. $E(S, T)$ stands for the set of edges from vertices in $S \subseteq V$ to vertices in $T \subseteq V$, that is

$$E(S, T) = \{(i, j) \in E : i \in S, j \in T\}$$

Then the density of subgraph induced by $S, T \subseteq V$ can be defined as

$$d(S,T) = \frac{|E(S,T)|}{\sqrt{|S||T|}}$$

The Densest Subgraph problem can be defined as

$$DS(G) = \max_{S,T \subseteq V} \{d(S,T)\}$$

1.3 Some Realistic Data Sets

The data sets that this application encounter can come from social networks (botnet followers) or e-commercial platforms (bully buyers). Thus the graph size can be huge. For example, the Twitter follower-followee data set used in Fraudar [15] contains 41.7 million nodes with 1.47 billion edges. Similar data sets for social networks such as Twitter and e-commercial platforms such as Amazon can be found on SNAP [21] or other platforms. It is intuitive that the size of graphs for such problem in real industry will be growing continuously. Hence it is important for the algorithms to have a linear or near linear run-time or be able to parallelize.

1.4 Dense Subgraph Detection-A Key Graph Kernel

Multiple algorithms exists for detecting the dense subgraphs. One majorly used algorithm is proposed by Charikar in 2000 [4], which is an approximation algorithm by greedy approach. Although Charikar's algorithm sacrificed quality of the result subgraph for much better time complexity, this algorithm still has a provable 2-approximation guarantee. That is, if the densest existing subgraph S' has edge density of $d(S') = \lambda$, the result subgraph S of Charikar's algorithm will have edge density of $d(S) \geq \lambda/2$.

The greedy idea of Charikar's algorithm is to remove the vertex that is least likely in the densest subgraph at each step according to certain rule. In the case of undirected graph, the rule can obviously be to remove the vertex with lowest degree. Then Charikar's algorithm can be described as following. [4]

procedure DENSEST-SUBGRAPH(G)
  Input: Undirected graph $G = (V, E)$.
  Output: Dense sugraph S of G.
  $n \leftarrow |V|$
  $G_n \leftarrow G$
  for $k \leftarrow n$ down to 1 **do**
    $v \leftarrow$ the vertex with smallest degree in G_k
    $G_{k-1} \leftarrow G_k \setminus \{v\}$
  return The subgraph with maximum density amoung $G_1, G_2, \ldots, G_n$.

A detailed proof for this algorithm to achieve a 2-approximation can be found in [19].

For directed graphs, Khuller and Saha proposed a approximation algorithm based on Charikar's algorithm in 2009 [19] that ultilized the same greedy idea. The key point of this algorithm for directed graphs is to first duplicate all the vertices and construct a bipartite graph such that one copy of the vertices have only outgoing edges and the other copy of the vertices have only incoming edges. Then the algorithm can be described as following. [19]

procedure DENSEST-SUBGRAPH-DIRECTED(G)

Input: Directed graph $G = (V, E)$.
Output: Dense sugraph S of G.
$n \leftarrow |V|$
$G_{2n} \leftarrow G$
for $k \leftarrow 2n$ down to 1 **do**
  $v \leftarrow$ the vertex with smallest degree in G_k
  if v has outgoing edges) **then**
    Delete all the outgoing edges incident on v.
  else
    Delete all the incoming edges incident on v.
  Deleted all vertices with 0 degree.
  $G_{k-1} \leftarrow$ the remaining of graph G_k
return The subgraph with maximum density amoung $G_1, G_2, \ldots, G_{2n}$.

A detailed proof for this algorithm to achieve a 2-approximation can also be found in [19].

Both algorithms has time complexity of $O(|V| \log |V|)$ and space complexity of $O(|V|^2|E|)$. To evaluate the performance of these two algorithms, both density on the result subgraph and runtime can be used.

1.5 Prior and Related Work

1.5.1 Dense Subgraph Problem

The history for Dense Subgraph problem for static graphs has a rather short history, as the best exact solution was proposed by Goldberg in 1984 [12] and the best approximation algorithm so far were proposed by Charikar in 2000 [4] and Khuller and Saha in 2009 [19] for undirected and directed graphs.

Goldberg's solution works only for undirected graph. His idea was to interestingly transfer this dense subgraph problem into a well-know min-cut problem by adding two vertices s and t. Both s and t are connected with all the vertices in graph $G = (V, E)$. For each vertex $v_i \in V$, edge (s, v_i) has edge weight that is the same as the degree of v_i and edge (v_i, t) has edge weight of a positive constant c. All the edges in the original graph has an edge weight of 1. Then buy performing a min-cut call that splits s and t into two subgraphs, one of the subgraphs would be the densest subgraph of G after removing s or t.

Since min-cut problem can be solved using the parametrix max-flow algotirhm, this algorithm has a $O(|V||E|)$ time complexity. Thus Goldberg's algorithm is not scalable for large graphs; faster approximation algorithms are more preferred in industry situations.

1.5.2 Fraud Detection

Data-driven approaches have received great success in the field of fraud detection [20, 15]: most methods indentify unexpected dense regions of the bipartite graph, as creating fake reviews/ratings unavoidably generates edges in the graph [16, 6, 23, 36, 29, 2].

Unexpected spectral patterns. Global graph mining methods model the entire graph to find fraud based on singular value decomposition (SVD), latent factor models, and belief propagation (BP). SpokEn [27] considered the "spokes" pattern produced by pairs of eigenvectors of graphs, and was later generalized for fraud detection. fBox [28] focuses on mini-scale attacks missed by spectral techniques. BP has been used for fraud classification on eBay [25], link farming on Twitter [9], and fake software review detection [1].

Unexpected high density in subgraphs. Finding dense subgraphs has been studied from a wide array of perspectives such as mining frequent subgraph patterns [22, 37], detecting communities [10, 5, 26], and finding quasi-cliques [11, 33, 8, 30]. [4] shows that average degree of subgraph can be maximized with approximation guarantees. [34] optimizes the density of adjacency matrix of subgraph with quality guarantees. [15] adopts both node degree and edge density to model suspiciousness of subgraph and further increases accuracy in binary adjacency matrix of bipartite graph.

Unexpected high density in time-series. Typically there are two kinds of representation on density in time-series. One is dense subgraphs in evolving graphs [7]. CopyCatch [3] uses local search heuristics to find Δt-bipartite cores in which users consistently likes the same Facebook pages at the same short time interval. The other is dense subtensors in high-order tensors of a time dimension [24, 17, 31] or tensor streams [32]. [35, 13] consider fraud detection methods that are robust to camouflage attacks. [14] adopts a Bayesian model to find early spikes of outlier ratings in time series. All these methods focus on the time-series domain, observing changes in the behavior from system access logs rather than graph data.

1.6 A Sequential Algorithm

Discuss here the outlines of a sequential algorithm. What programming paradigms might make the most sense? What are the key data structures? Does the computational complexity differ from that in the Section 1.4?

1.7 A Reference Sequential Implementation

Discuss here your implementation of the basic sequential code. Include what language/paradigm you used for the code.

1.8 Sequential Scaling Results

Discuss here results from your sequential implementation. Include software and hardware configuration, where the input graph data sets came from, and how input data set characteristics were varied. Did the performance as a function of size vary as you predicted?

1.9 An Enhanced Algorithm

Discuss here the outlines of an enhanced algorithm. This could be a parallel code, a code with some significant heuristics, or a code written in a non-traditional programming paradigm. Pseudocode is fine. Discuss what you think is the computational complexity.

1.10 A Reference Enhanced Implementation

Discuss here an implementation of the enhanced algorithm. Include what language/paradigm you used for the code.

1.11 Enhanced Scaling Results

Discuss here results from the enhanced algorithm. Include software and hardware configuration, where the input graph data sets came from, and how input data set characteristics were varied. Ideally plots of performance vs BOTH problem size changes AND hardware resources are desired. Did the performance as a function of size vary as you predicted?

1.12 Conclusion

Summarize your paper. Discuss possible future work and/or other options that may make sense.

1.13 Response to Reviews

This will be included only in the second and third iterations, and will be a summary of what you learned from the reviews you received from the prior pass, and how you modified the paper accordingly.

Bibliography

[1] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. In *WSDM*, pages 2–11, 2013.

[2] Yikun Ban, Xin Liu, Tianyi Zhang, Ling Huang, Yitao Duan, Xue Liu, and Wei Xu. Badlink: Combining graph and information-theoretical features for online fraud group detection. *arXiv preprint arXiv:1805.10053*, 2018.

[3] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, pages 119–130, 2013.

[4] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 84–95. Springer, 2000.

[5] Jie Chen and Yousef Saad. Dense subgraph extraction with application to community detection. *IEEE TKDE*, 24(7):1216–1230, 2012.

[6] Carter Chiu, Justin Zhan, and Felix Zhan. Uncovering suspicious activity from partially paired and incomplete multimodal data. *IEEE Access*, 5:13689–13698, 2017.

[7] Alessandro Epasto, Silvio Lattanzi, and Mauro Sozio. Efficient densest subgraph computation in evolving graphs. In *WWW*, pages 300–310, 2015.

[8] Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. Top-k overlapping densest subgraphs. *DMKD*, 30(5):1134–1165, 2016.

[9] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *WWW*, pages 61–70, 2012.

[10] Christos Giatsidis, Dimitrios M Thilikos, and Michalis Vazirgiannis. D-cores: Measuring collaboration of directed graphs based on degeneracy. In *ICDM*, pages 201–210, 2011.

[11] Christos Giatsidis, Dimitrios M Thilikos, and Michalis Vazirgiannis. Evaluating cooperation in communities with the k-core structure. In *ASONAM*, pages 87–93, 2011.

[12] Andrew V Goldberg. *Finding a maximum density subgraph*. University of California Berkeley, CA, 1984.

[13] Zhongshu Gu, Kexin Pei, Qifan Wang, Luo Si, Xiangyu Zhang, and Dongyan Xu. Leaps: Detecting camouflaged attacks with statistical learning guided by program analysis. In *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*, pages 57–68. IEEE, 2015.

[14] Bryan Hooi, Neil Shah, Alex Beutel, Stephan Günnemann, Leman Akoglu, Mohit Kumar, Disha Makhija, and Christos Faloutsos. Birdnest: Bayesian inference for ratings-fraud detection. In *SDM*, pages 495–503, 2016.

[15] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD*, pages 895–904, 2016.

[16] Xia Hu, Jiliang Tang, and Huan Liu. Online social spammer detection. In *AAAI*, volume 14, pages 59–65, 2014.

[17] Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. Spotting suspicious behaviors in multimodal data: A general metric and algorithms. *IEEE TKDE*, 28(8):2187–2200, 2016.

[18] Ravi Kannan and V Vinay. *Analyzing the structure of large graphs*. Rheinische Friedrich-Wilhelms-Universität Bonn Bonn, 1999.

[19] Samir Khuller and Barna Saha. On finding dense subgraphs. In *International Colloquium on Automata, Languages, and Programming*, pages 597–608. Springer, 2009.

[20] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *WWW*, pages 933–943, 2018.

[21] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[22] Chao Liu, Xifeng Yan, Hwanjo Yu, Jiawei Han, and Philip S Yu. Mining behavior graphs for "backtrace" of noncrashing bugs. In *SDM*, pages 286–297, 2005.

[23] Shenghua Liu, Bryan Hooi, and Christos Faloutsos. Holoscope: Topology-and-spike aware fraud detection. In *CIKM*, pages 1539–1548, 2017.

[24] Koji Maruhashi, Fan Guo, and Christos Faloutsos. Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis. In *ASONAM*, pages 203–210, 2011.

[25] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, pages 201–210, 2007.

[26] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. Focused clustering and outlier detection in large attributed graphs. In *KDD*, pages 1346–1355, 2014.

[27] B Prakash, Ashwin Sridharan, Mukund Seshadri, Sridhar Machiraju, and Christos Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. *Advances in knowledge discovery and data mining*, pages 435–448, 2010.

[28] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In *ICDM*, pages 959–964, 2014.

[29] Hua Shen, Fenglong Ma, Xianchao Zhang, Linlin Zong, Xinyue Liu, and Wenxin Liang. Discovering social spammers from multiple views. *Neurocomputing*, 225:49–57, 2017.

[30] Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. Corescope: Graph mining using k-core analysispatterns, anomalies and algorithms. In *ICDM*, pages 469–478, 2016.

[31] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. D-cube: Dense-block detection in terabyte-scale tensors. In *WSDM*, pages 681–689, 2017.

[32] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. Densealert: Incremental dense-subtensor detection in tensor streams. In *KDD*, 2017.

[33] Charalampos Tsourakakis. The k-clique densest subgraph problem. In *WWW*, pages 1122–1132, 2015.

[34] Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *KDD*, pages 104–112, 2013.

[35] Sankar Virdhagriswaran and Gordon Dakin. Camouflaged fraud detection in domains with complex relationships. In *KDD*, pages 941–947, 2006.

[36] Soroush Vosoughi, MostafaNeo Mohsenvand, and Deb Roy. Rumor gauge: predicting the veracity of rumors on twitter. *ACM TKDD*, 11(4):50, 2017.

[37] Zhaonian Zou, Jianzhong Li, Hong Gao, and Shuo Zhang. Mining frequent subgraph patterns from uncertain graph data. *IEEE TKDE*, 22(9):1203–1218, 2010.