Capitalizing on Hierarchical Graph Decomposition for Scalable Network Analysis

Rakhi Saxena
Supervised by Sharanjit Kaur and Vasudha Bhatnagar
University of Delhi at New Delhi, India
rsaxena@db.du.ac.in

ABSTRACT

Processing large graphs has become commonplace across many academic and industrial applications. We address the computational challenge of analyzing large networks on a single consumer-grade machine. Our strategy involves arranging networks into layers of smaller, increasingly cohesive subgraphs, which is motivated by the observation that real-world networks exhibit a hierarchical organization. We decompose large networks to reveal the underlying hierarchy and extract signals from this hierarchical topology to solve three network analysis problems viz., network comparison, determining influential spreaders, and centrality computation. Empirical investigation reveals that our approach is effective and faster than state-of-the-art competing algorithms.

1. INTRODUCTION

Complex networks have attracted immense attention because of their ability to model a wide variety of associations between entities in social networks, power-grids, transportation, biological systems etc. Many of these networks contain millions of nodes and billions of edges, and the data is easily available on the world wide web. Several distributed as well as disk-based frameworks [3] have been specifically designed to analyze such large networks. However, developing algorithms for these frameworks requires expertise in the use of highly specialized programming paradigms.

In this study, we explore the question: Is graph decomposition a viable strategy for effective network analysis using a single consumer-grade machine? We address this question by decomposing graphs into increasingly cohesive parts using two well-known hierarchical graph decomposition algorithms [6, 17]. The choice of this strategy is emboldened by the fact that it is viable to compute graph decomposition for networks of billions of edges on a consumer-grade PC [9]. Our approach is simple, intuitive and motivated by the observation that real-world networks from a wide variety of domains have an inherently hierarchical organization [13]. Hierarchy has been recognized as a critical organizational property for many complex systems, ranging from biological networks, societies, to road networks and the Internet [7]. Massive networks of the current technology era have been analyzed, visualized and understood better after hierarchical decomposition [1, 10].

My Ph.D. dissertation focuses on developing effective and fast algorithms for network analysis using a single PC by leveraging signals acquired from hierarchically decomposed networks. The approach is to decompose the given graph into a nested hierarchy of increasingly cohesive subgraphs. The hierarchy imparts a natural ordering to the vertices, and the cohesive regions revealed by the decomposition mimic community structures.

We leverage hierarchy and approximated community structure to develop algorithms for three common problems in network analysis, namely, network comparison, finding influential spreaders and computing node centrality. The effectiveness of these algorithms establishes that i) hierarchy in networks is a potent property for network discrimination ii) links between levels of the hierarchy are a fair approximation of intra- and inter-community ties iii) vertices at the same level have similar importance and similar capability to diffuse information. In summary, we find that hierarchical decomposition of networks is a meritorious approach for three network analysis tasks.

Organization: Sec. 2 introduces two graph decomposition methods. Sec. 3 presents three algorithms for network analysis. Sec. 4 delineates directions for future research.

2. HIERARCHICAL GRAPH DECOMPOSITION METHODS

In this section, we introduce k-core [17] and k-truss [6] decomposition methods that we use for eliciting network hierarchy.

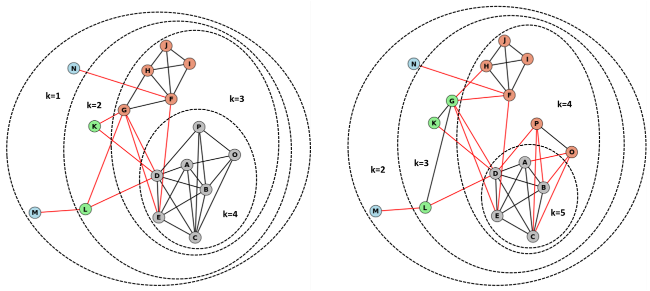
Let $G = (V, E)$ be a simple, connected, undirected graph, where V represents the set of vertices and $E \subseteq V \times V$ represents the set of edges[1]. An edge $e_{ij} \in E$ iff it connects vertices $v_i, v_j \in V$. Set $N_i = \{v_j \in V | e_{ij} \in E\}$ denotes the set of neighbours of vertex v_i. The degree of a vertex $\delta_i = |N_i|$ denotes the number of neighbours of v_i.

2.1 k-core Decomposition

The k-core decomposition organizes the graph into a hierarchy of subgraphs (called k-cores) such that the degree of

Proceedings of the VLDB 2018 Ph.D. Workshop, August 27, 2018. Rio de Janeiro, Brazil.

[1] We use terms network/graph, node/vertex, and edge/link interchangeably.



(a) *k-core Decomposition* (b) *k-truss Decomposition*

Figure 1: Hierarchical decomposition of a toy network. Nodes with the same coreness/trussness have the same color. Edges colored red connect nodes in different hierarchy layers.

every vertex in a k-core is at least k. A vertex that belongs to a k-core but not to a k+1 core has coreness k. Definitions adapted from [17] follow.

Definition 2.1. *A subgraph, $C_k = (V_k, E_k|V_k)$ of G is a k-core iff $\forall v_i \in V_k : \delta_i >= k$ and C_k is the maximal subgraph with this property.* □

Definition 2.2. *Coreness (κ_i) of vertex v_i is k i.e. $\kappa_i = k$ iff $v_i \in C_k \wedge v_i \notin C_{k+1}$.* □

Fig. 1a illustrates the k-core decomposition of a toy network with 16 nodes and 34 edges. We use an efficient $O(|E|)$ k-core decomposition algorithm as proposed in [2].

2.2 k-truss Decomposition

The k-truss decomposition organizes a given graph into a hierarchy of subgraphs (called k-trusses) such that every edge in a k-truss is part of at least $(k-2)$ triangles [6]. An edge that belongs to a k-truss but not to a (k+1)-truss has trussness k. Definitions adapted from [18] follow.

Definition 2.3. *Support (σ_{ij}) of edge e_{ij} is $|N_i \cap N_j|$* □

Definition 2.4. *Subgraph, $T_k = (V_k, E_k|V_k)$ of G is a k-truss iff $\forall e_{ij} \in E_k : \sigma_{ij} >= (k-2)$, and T_k is the maximal subgraph with this property.* □

Definition 2.5. *Trussness $(\top_{ij})$ of edge e_{ij} is k i.e. $\top_{ij} = k$ iff $e_{ij} \in T_k \wedge e_{ij} \notin T_{k+1}$* □

We define trussness of nodes in the graph. A node that belongs to the k-truss but not to the (k+1)-truss has trussness k. Formally,

Definition 2.6. *Trussness (τ_i) of node v_i is k iff $v_i \in T_k \wedge v_i \notin T_{k+1}$* □

Fig. 1b illustrates the k-truss decomposition of a toy network. We use the elegant in-memory $O(m^{1.5})$ k-truss decomposition algorithm proposed in [18] so that large network decomposition is feasible on a consumer-grade machine.

3. NETWORK ANALYSIS ALGORITHMS

Next, we outline the solutions to the three network analysis tasks mentioned in Sec. 1.

Algorithm	Purity	Precision	Recall	Accuracy	NMI
NSD-C	1.0	1.0	1.0	1.0	1.0
NSD-T	1.0	1.0	1.0	1.0	1.0
NCKD	0.67	0.51	0.87	0.73	0.67

Table 1: Quality metrics for hierarchical clustering of 15 large real-world networks.

3.1 Network Comparison

The task of network comparison is encountered in several domains such as pattern recognition, analyses of the functionality of biological networks, and study of the temporal evolution of networks. Graph comparison entails computation of distance between a pair of networks to measure the extent of similarity between them. State-of-the-art network comparison methods extract network features and the distance between features quantifies network similarity.

Existing algorithms make use of features extracted from either local neighborhood of vertices [4, 20] or global network topology [12]. Both k-core and k-truss algorithms promote local (node/edge level) features to obtain global (graph level) feature (i.e. hierarchy) of the network, which forms the basis of characterizing networks. In this way our approach plugs the gap between the use of myopic local features, and non-scalable global features.

We first experimented with network signatures extracted from simple but efficient features from k-core decomposition [16]. Encouraged by the results, we extended the study to improve effectiveness by using sophisticated aggregation of features derived from k-core and k-truss decompositions [14].

Algorithm NCKD uses probability distribution of nodes at each level of the hierarchy, and the distribution of edges within the level and between different levels [16]. Jensen-Shannon distance between two probability distributions extracted from a given pair of networks quantifies structural differences between them. Empirical investigations establish that these two distributions are capable of capturing structural differences between networks and discriminating between different genres of networks reasonably well.

Next, we use more expressive methods of distribution aggregation and additionally examine truss based decomposition for extracting network signatures [14]. NSD-C and NSD-T algorithms examine core-based and truss-based decomposition respectively, to asses node-level assortativity (propensity to connect with other nodes at the same level) in addition to hierarchy levels of the nodes. Quantiles of the distributions of the two features are used as network signature and Canberra distance between signatures is computed.

In order to evaluate, fifteen large public real-world networks[2] from three genres were clustered with the assumption that graphs belonging to the same genre are structurally more similar and hence should be grouped together by an effective network comparison algorithm. Table 1 reports quality metrics of the resultant clustering scheme delivered by our algorithms. Having found the performance of the three algorithms better than the state-of-the-art algorithms, we compared NSD-C, NSD-T and NCKD for accuracy, speed and sensitivity to noise and missing data. Further experiments lead to the following conclusions. Network comparison using i) network signatures based on simple probability distributions of hierarchy levels of nodes and edges is the

[2] Refer to paper [16] for details of the networks.

fastest, but relatively least accurate, ii) quantile-based aggregation of core levels of nodes and their assortativity is the best of the three measures, but is less sensitive to noise and missing data, iii) quantile-based aggregation of truss levels of nodes and their assortativity is the slowest of the three methods, but most sensitive to noise and missing data.

3.2 Influential Spreader

Predicting individuals who influence the spread of information is another important task in social network analysis. Prerequisite for understanding the spreading dynamics in online social networks, the task also finds applications in product marketing, promotion of innovative ideas, restricting negative information etc.. State-of-the-art methods for predicting influential actors use facets such as - strength of interaction with neighbors [1], community structure [19], or hierarchy [10] in the network.

These methods for finding influential spreaders miss out on the advantage of the interplay of the three facets. We address the research gap by exploiting the synergy between the three facets, and demonstrate significant improvement over existing methods for prediction of influential spreaders.

The proposed influence scoring method IPRI (Influence scoring using Position, Reachability, and Interaction) [8] uses i) position of the actor in the network hierarchy, ii) intensity of his interactions with neighbors and iii) extent of actor's connectivity in different communities. The algorithm uses k-truss decomposition method, which confers the dual advantage of revealing hierarchy and homophilic groups (approximate communities) in the network. IPRI algorithm computes the following three indices for every vertex:

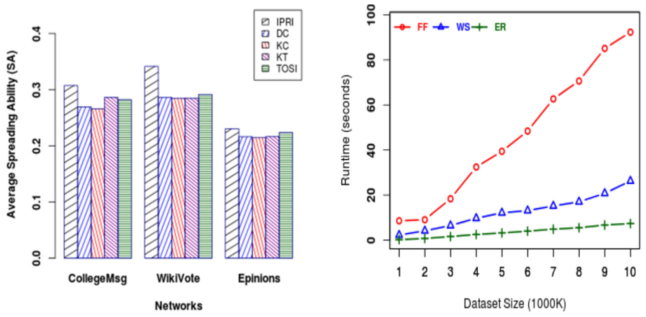
i) Positional Index (τ): Trussness of a node obtained by decomposition of the network proxies for its relative position in network hierarchy. A higher level indicates larger neighborhood span that aids wider spread of information. Positional Index of node v_i is same as its trussness τ_i.

ii) Reachability Index (ρ): A node having connections with more truss levels has higher reachability in terms of information propagation, compared to a node having connections with fewer truss levels [19]. We quantify a node's reachability to diverse communities as the entropy of trussness of its neighbors.

iii) Interaction Index (μ): The propagation of information is governed by the strength of interaction not only with neighbors but also with 2-steps neighbors [11][3]. Based on this observation, the interaction index of a node is computed as the sum of the sum of weights of edges incident on neighbors scaled by their respective positional index.

IPRI algorithm computes the influence score by integrating positional index, reachability index and interaction index of a node using a multiplicative function. The score is indicator of the power to influence other users in the network, with higher score indicating more influence.

Experimentation with large real-world social networks establishes the validity and accuracy of the scoring method. We evaluate effectiveness of competing methods by using SIR epidemic model on three large real-world networks (CollegeMsg, WikiVote, and Epinions)[4]. Performance of IPRI is compared with four measures - degree centrality (DC), k-core (KC), k-truss (KT), Trust-Oriented Social Influencers (TOSI). For each competing measure, top 20% nodes are taken as initial spreaders and 100 simulations of SIR model are run to capture the average spreading ability (SA) of top-rankers. SA of the initial set of infected nodes is quantified as the percentage of nodes infected during spreading process. Figure 2a shows that average SA of IPRI algorithm is higher than that of competing measures for all networks afffirming its better effectiveness.



(a) *Spreading Ability of IPRI*

(b) *SC execution time averaged over 10 runs*

Figure 2: **Results of evaluation of IPRI and SC.**

3.3 Social Centrality

Centrality is widely-used for identifying important nodes in a network [5]. Existing methods for discovering important nodes do not take cognizance of inherent hierarchy and community structure in human-centric networks for determining centrality of actors. Since humans derive benefits concomitant with their position in the network hierarchy, and with the strength of their intra– and inter–community connections, we posit that a centrality measure that takes these aspects into account gauges the importance of individuals more realistically in human-centric networks.

Based on the mature theory of *social capital*, the proposed Social Centrality score (SC) emulates the real-life behavior of social actors to bond within community and bridge between communities [15]. SC score of a node quantifies its ability to mobilize resources in the network based on its location in the hierarchy, embeddedness in community and intensity of relations with neighbors. Application of k-truss decomposition elicits network hierarchy and approximated community structure. We posit that if two nodes and the connecting edge all have the same trussness, they are part of the same community, and the connecting edge is an intra-community edge. SC score is computed by extracting following three nodal properties from the hierarchical decomposition of a graph.

i) Sociability Index (ω): Sociability quantifies the extent to which an actor can leverage the resources controlled by its immediate neighbors. It takes into consideration size of the immediate neighborhood (ego-network) and intensity of relationships (edge-weights). Sociability index of a node is defined as sum of weights of edges incident on it.

ii) Bonding Potential (β): Bonding potential of an individual is determined by his hierarchical position in G as well as by the sociability of his intra-community neighbors.

[3] Liu et al. [11] establish that the 2-step neighborhood of nodes is a good choice that balances cost and performance when identifying influencers.

[4] Refer to paper [8] for details of the networks.

Bonding potential of an actor is quantified as the sum of sociability index of its intra-community neighbors weighted by his position in the hierarchy.

iii) Bridging Potential (γ): An actor with links in diverse communities can draw advantages that are not available within his community. The bridging potential of an actor is the sum of the intensity of relationship with the inter-community neighbors scaled by their respective positions in the hierarchy.

SC score of a node is computed by aggregating its sociability index, bonding and bridging potential using a multiplicative function. The empirical study based on diverse, human-centric networks vindicates the propositional basis of the model and demonstrates its validity, effectiveness, and superior performance compared to prevailing centrality measures. We also performed scalability tests on synthetic networks generated from Erdös-Rényi (ER), Watts-Strogatz (WS) and Forest Fire (FF) models with the number of nodes varying from 1 million to 10 million and edges ranging from 2 million to ≈60 million. The results (Figure 2b) endorse our claim of the ability of hierarchy-based algorithms to process large graphs on consumer-grade PC.

4. CONCLUSION AND FUTURE WORK

In this doctoral study, we find that large networks can be analyzed effectively on a single consumer-grade machine after hierarchical decomposition. Specifically, we explored three network analysis problems - network comparison, identifying influence spreaders and computing centrality in human-centric social networks. Existing efficient algorithms for k-core and k-truss decomposition are fundamental to our solutions. The hierarchy of nodes and links between levels of the hierarchy are reasonably effective signals to quantify network similarity. We also observe that the cohesive regions of the network revealed by the decomposition make a good approximation of community structure. Integrating hierarchy, and the resulting approximate community structure is superior to the state-of-the-art algorithms for computing centrality and identifying influential spreaders.

The study opens few new questions - What other network analysis tasks can be solved effectively and efficiently for massive graphs using hierarchical graph decomposition methods? Is the divide-and-conquer approach in general, practical for analysis of large graphs? Can horizontal partitioning methods be leveraged for designing scalable algorithms? Answering these questions demands intense involvement in the theoretical underpinnings of graph decomposition methods.

5. REFERENCES

[1] M. A. Al-garadi, K. D. Varathan, and S. D. Ravana. Identification of Influential Spreaders in Online Social Networks using Interaction Weighted K-core Decomposition Method. *Physica A*, 468(C):278–288, 2017.

[2] V. Batagelj and M. Zaveršnik. Fast Algorithms for Determining (Generalized) Core Groups in Social Networks. *Advances in Data Analysis and Classification*, 5(2):129–145, Jul 2011.

[3] O. Batarfi, R. E. Shawi, A. G. Fayoumi, R. Nouri, S. Beheshti, A. Barnawi, and S. Sakr. Large Scale Graph Processing Systems: Survey and an Experimental Evaluation. *Cluster Computing*, 18(3):1189–1213, Sep 2015.

[4] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos. Network Similarity via Multiple Social Theories. In *Proceedings of IEEE/ACM ASONAM*, pages 1439–1440, 2013.

[5] F. Bloch, M. O. Jackson, and P. Tebaldi. Centrality Measures in Networks. *CoRR*, abs/1608.05845, 2016.

[6] J. Cohen. Trusses: Cohesive Subgraphs for Social Network Analysis. *NSA:Technical report*, 2008.

[7] K. D. Farnsworth, L. Albantakis, and T. Caruso. Unifying Concepts of Biological Function from Molecules to Ecosystems. *Oikos*, 126(10):1367–1376, 2017.

[8] S. Kaur, R. Saxena, and V. Bhatnagar. Leveraging Hierarchy and Community Structure for Determining Influencers in Networks. In *Proceedings of DaWaK*, pages 383–390, 2017.

[9] W. Khaouid, M. Barsky, V. Srinivasan, and A. Thomo. K-Core Decomposition of Large Networks on a Single PC. *PVLDB*, 9(1):13–23, 2015.

[10] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse. Identification of Influential Spreaders in Complex Networks. *Nature Physics*, 6(11):888–893, Aug 2010.

[11] Y. Liu, M. Tang, T. Zhou, and Y. Do. Identify Influential Spreaders in Complex Networks, the Role of Neighborhood. *Physica A: Statistical Mechanics and its Applications*, 452:289–298, 2016.

[12] S. Lu, J. Kang, W. Gong, and D. Towsley. Complex Network Comparison using Random Walks. In *Proceedings on WWW*, pages 727–730, 2014.

[13] H. Mengistu, J. Huizinga, J. Mouret, and J. Clune. The Evolutionary Origins of Hierarchy. *PLoS Computational Biology*, 12(6):e1004829, 2016.

[14] R. Saxena, S. Kaur, and V. Bhatnagar. Identifying Similar Networks using Structural Hierarchy. *Ready for Submission; Avaialable on Request.*

[15] R. Saxena, S. Kaur, and V. Bhatnagar. Social Centrality using Network Hierarchy and Community Structure. *Data Mining and Knowledge Discovery*, Accepted for publicaton, 2018.

[16] R. Saxena, S. Kaur, D. Dash, and V. Bhatnagar. Leveraging Structural Hierarchy for Scalable Network Comparison. In *Proceedings of DEXA*, pages 287–302, 2016.

[17] S. B. Seidman. Network Structure and Minimum Degree. *Social Networks*, 5:269–287, 1983.

[18] J. Wang and J. Cheng. Truss Decomposition in Massive Networks. *Proceedings of the VLDB Endowment*, 5(9):812–823, 2012.

[19] S. Wang, F. Wang, Y. Chen, C. Liu, Z. Li, and X. Zhang. Exploiting Social Circle Broadness for Influential Spreaders Identification in Social Networks. *World Wide Web*, 18(3):681–705, 2015.

[20] A. E. Wegner, L. Ospina-Forero, R. E. Gaunt, C. M. Deane, and G. Reinert. Identifying Networks with Common Organizational Principles. *Journal of Complex Networks*, 2018.