Tutorial

Core Decomposition in Graphs: Concepts, Algorithms and Applications

Fragkiskos D. Malliaros[1], Apostolos N. Papadopoulos[2], Michalis Vazirgiannis[1]
[1]Computer Science Laboratory, École Polytechnique, France
[2]Department of Informatics, Aristotle University of Thessaloniki, Greece
{fmalliaros, mvazirg}@lix.polytechnique.fr, papadopo@csd.auth.gr

ABSTRACT

Graph mining is an important research area with a plethora of practical applications. Core decomposition in networks, is a fundamental operation strongly related to more complex mining tasks such as community detection, dense subgraph discovery, identification of influential nodes, network visualization, text mining, just to name a few. In this tutorial, we present in detail the concept and properties of core decomposition in graphs, the associated algorithms for its efficient computation and some of its most important applications.

1. INTRODUCTION

Core decomposition is a well-studied topic in graph mining. Informally, the k-core decomposition is a threshold-based hierarchical decomposition of a graph into nested subgraphs. The basic idea is that a threshold k is set on the degree of each node; nodes that do not satisfy the threshold, are excluded from the process. There exists a rich literature studying algorithmic aspects of core decomposition by taking different viewpoints, such as distributed, streaming, disk-resident data, to name a few. In addition, core decomposition has been used successfully in many diverse application domains, including social networks analysis and text analytics tasks.

Next, we formally define the concept of k-core decomposition in graphs. Let $G = (V, E)$ be an undirected graph. Let H be a subgraph of G, i.e., $H \subseteq G$. Subgraph H is defined to be a *k-core* of G, denoted by G_k, if it is a maximal connected subgraph of G in which all nodes have degree at least k. The *degeneracy* $\delta^*(G)$ of a graph G is defined as the maximum k for which graph G contains a non-empty k-core subgraph. A node i has *core number* $c_i = k$, if it belongs to a k-core but not to any $(k+1)$-core. The *k-shell* is the subgraph defined by the nodes that belong to the k-core but not to the $(k+1)$-core.

Based on the above definitions, it is evident that if all the nodes of the graph have degree at least one, i.e., $d_v \geq 1, \forall v \in V$, then the 1-core subgraph corresponds to the whole graph, i.e., $G_1 \equiv G$. Furthermore, assuming that $G_i,\ i = 0, 1, 2, \ldots, \delta^*(G)$ is the i-core of G, then the k-core subgraphs are nested, i.e., $G_0 \supseteq G_1 \supseteq G_2 \supseteq \ldots \supseteq G_{\delta^*(G)}$. Typically, subgraph $G_{\delta^*(G)}$ is called *maximal k-core subgraph* of G.
Figure 1 depicts an example of a graph and the corresponding k-core decomposition. As we observe, the degeneracy of this graph is $\delta^*(G) = 3$; thus, the decomposition creates three nested k-core subgraphs, with the 3-core being the maximal one. The nested structure of the k-core subgraphs is indicated by the dashed lines. Furthermore, the color on the nodes indicates the core number c of each node. Lastly, we should note here that the k-core subgraphs are not necessarily connected.

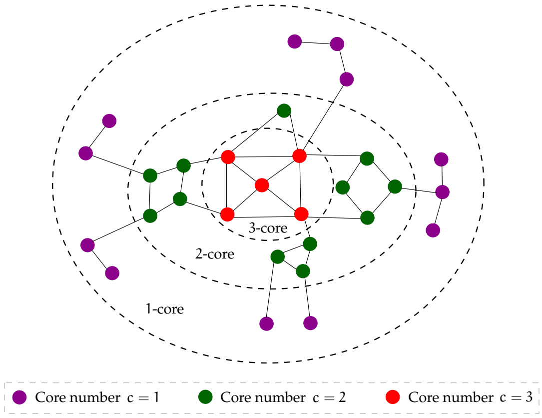

Figure 1: Example of the k-core decomposition.

2. GOALS AND OUTLINE

The goal of this tutorial is to present in detail the algorithmic paradigm of core decomposition in graphs. In particular, we will focus on the following points:

(i) **Fundamental concepts of core decomposition.** We present the notion of k-core decomposition for unweighted and undirected graphs and then extensions for weighted, directed, probabilistic and signed ones. We also present generalizations of the decomposition to node properties beyond the degree.

(ii) **Algorithms for core decomposition.** Computing the k-core decomposition of a graph can be done through a simple process that is based on the following property: to extract the k-core subgraph, all nodes with degree less than k and their adjacent edges should be recursively deleted. In the tutorial, we present efficient algorithms for the k-core decomposition. We also examine several extensions that have been proposed by the databases community for large scale k-core decomposition under various computation frameworks, including streaming, distributed and disk-based algorithms. We also examine how to estimate the k-core number of each node using only local information.

(iii) **Applications.** We demonstrate applications of the k-core decomposition in various domains, including dense subgraph detection, graph clustering, modeling of network dynamics and network visualization.

The outline of the tutorial has as follows:

1. Introduction

– Social network analysis
– Highlights of core decomposition

2. Fundamental Concepts of Core Decomposition

– k-core subgraph, k-shell subgraph, k-core number, degeneracy
– Weighted networks, directed networks, signed networks, probabilistic networks
– Generalized cores
– Truss decomposition
– Extensions of the core decomposition

3. Algorithms

– Baseline algorithm
– An $\mathcal{O}(|E|)$ algorithm for k-core decomposition
– Streaming k-core decomposition
– Distributed k-core decomposition
– Disk-based k-core decomposition
– Local estimation of k-core numbers

4. Applications in Complex Networks

– Dense subgraph discovery
– Community detection and evaluation
– Identification of influential nodes
– Dynamics of networks
– Modeling the Internet topology
– Network visualization
– Text mining

5. Open Problems and Future Research

– Algorithms and applications

3. ACKNOWLEDGEMENTS

Fragkiskos D. Malliaros is a recipient of the Google Europe Fellowship in Graph Mining, and this research is supported in part by this Google Fellowship.

4. REFERENCES

[1] J. I. Alvarez-Hamelin, A. Barrat, and A. Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In *NIPS*, pages 41–50, 2006.

[2] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. k-core decomposition of internet graphs: Hierarchies, self-similarity and measurement biases. *NHM*, 3(2):371, 2008.

[3] R. Andersen and K. Chellapilla. Finding dense subgraphs with size bounds. In *WAW*, pages 25–37, 2009.

[4] V. Batagelj and M. Zaversnik. Generalized cores. *CoRR*, 2002.

[5] V. Batagelj and M. Zaversnik. An $\mathcal{O}(m)$ algorithm for cores decomposition of networks. *CoRR*, 2003.

[6] F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. Core decomposition of uncertain graphs. In *KDD*, pages 1316–1325, 2014.

[7] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir. A model of internet topology using k-shell decomposition. *PNAS*, 104(27):11150–11154, 2007.

[8] J. Cheng, Y. Ke, S. Chu, and M. T. Ozsu. Efficient core decomposition in massive networks. In *ICDE*, pages 51–62, 2011.

[9] A. Garas, F. Schweitzer, and S. Havlin. A k-shell decomposition method for weighted networks. *New Journal of Physics*, 14(8), 2012.

[10] C. Giatsidis, F. D. Malliaros, D. M. Thilikos, and M. Vazirgiannis. CoreCluster: A degeneracy based graph clustering framework. In *AAAI*, pages 44–50, 2014.

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

[12] C. Giatsidis, D. M. Thilikos, and M. Vazirgiannis. D-cores: measuring collaboration of directed graphs based on degeneracy. *Knowl. Inf. Syst.*, 35(2):311–343, 2013.

[13] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljerosand, L. Muchnik, H. E. Stanley, and H. A. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 2010.

[14] R.-H. Li, L. Qin, J. X. Yu, and R. Mao. Influential community search in large networks. *Proc. VLDB Endow.*, 8(5):509–520, 2015.

[15] F. D. Malliaros, M.-E. G. Rossi, and M. Vazirgiannis. Locating influential nodes in complex networks. *Sci. Rep.*, 2016.

[16] F. D. Malliaros and M. Vazirgiannis. To stay or not to stay: modeling engagement dynamics in social graphs. In *CIKM*, pages 469–478, 2013.

[17] F. D. Malliaros and M. Vazirgiannis. Vulnerability assessment in social networks under cascade-based node departures. *EPL (Europhysics Letters)*, 110(6):68006, 2015.

[18] A. Montresor, F. D. Pellegrini, and D. Miorandi. Distributed k-core decomposition. *IEEE Transactions on Parallel and Distributed Systems*, 24(2):288–300, 2013.

[19] M. P. O'Brien and B. D. Sullivan. Locally estimating core numbers. In *ICDM*, pages 460–469, 2014.

[20] K. Pechlivanidou, D. Katsaros, and L. Tassiulas. MapReduce-based distributed K-shell decomposition for online social networks. In *SERVICES*, pages 30–37, 2014.

[21] M.-E. G. Rossi, F. D. Malliaros, and M. Vazirgiannis. Spread it good, spread it fast: Identification of influential nodes in social networks. In *WWW*, pages 101–102, 2015.

[22] A. E. Saríyüce, B. Gedik, G. Jacques-Silva, K.-L. Wu, and U. V. Çatalyürek. Streaming algorithms for k-core decomposition. *Proc. VLDB Endow.*, 6(6):433–444, Apr. 2013.

[23] A. E. Sariyuce, C. Seshadhri, A. Pinar, and U. V. Catalyurek. Finding the hierarchy of dense subgraphs using nucleus decompositions. In *WWW*, pages 927–937, 2015.

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

[25] N. Tatti and A. Gionis. Density-friendly graph decomposition. In *WWW*, pages 1089–1099, 2015.

[26] E. Valari, M. Kontaki, and A. N. Papadopoulos. Discovery of top-k dense subgraphs in dynamic graph collections. In *SSDBM*, pages 213–230, 2012.

[27] J. Wang and J. Cheng. Truss decomposition in massive networks. *Proc. VLDB Endow.*, 5(9):812–823, 2012.

[28] G.-Q. Zhang, G.-Q. Zhang, Q.-F. Yang, S.-Q. Cheng, and T. Zhou. Evolution of the Internet and its cores. *New Journal of Physics*, 10(12):123027+, 2008.

[29] Y. Zhang and S. Parthasarathy. Extracting analyzing and visualizing triangle k-core motifs within networks. In *ICDE*, pages 1049–1060, 2012.