Leveraging Structural Hierarchy for Scalable Network Comparison

Rakhi Saxena[1], Sharanjit Kaur[2](✉), Debasis Dash[3], and Vasudha Bhatnagar[4]

[1] Deshbandhu College, University of Delhi, Delhi, India
rsaxena@db.du.ac.in
[2] Acharya Narendra Dev College, University of Delhi, Delhi, India
sharanjitkaur@andc.du.ac.in
[3] CSIR-Institute of Genomics and Integrative Biology, Delhi, India
ddash@igib.res.in
[4] Department of Computer Science, University of Delhi, Delhi, India
vbhatnagar@cs.du.ac.in

Abstract. *K-core* decomposition is a popular method that segments a network revealing the underlying hierarchy. We explore the propensity of this decomposition method for structural discrimination among networks by extracting features from each level of the hierarchy. We propose a novel algorithm for *N*etwork *C*omparison using *k-core D*ecomposition (*NCKD*). The method is effective, efficient and scalable, with computational complexity of $O(|\mathcal{E}|)$, where $\mathcal{E}$ is the set of edges in the network. The low computational complexity of the method makes it attractive for scalable network comparison.

NCKD algorithm decomposes networks and extracts features from the resulting shells. Jensen-Shannon distance between extracted features quantifies structural differences between networks. We establish that probability distributions of coreness and intra/inter-shell edges are capable of characterizing different genres of networks and capturing finer structural differences between networks of the same genre. We experiment with synthetic and real-life networks up to eight million edges on a single PC. Comparison with two recent state-of-the-art network comparison methods affirms that *NCKD* outperforms in terms of effectiveness and scalability.

Keywords: Network comparison · K-core decomposition · Graph analytics · Social networks · Jensen-Shannon distance

1 Introduction

Complex networks have attracted immense attention because of their ability to model social relations, power grids, transportation links, biological processes etc. [7]. One of the challenging tasks in network analytics is to assess and quantify similarity between two networks. Applications of network comparison include construction of phylogenetic trees and function prediction in biological networks, studying evolution in social networks, analysing semantic structure in natural languages, detecting code theft by comparing two executable objects etc. [7,9].

S. Hartmann and H. Ma (Eds.): DEXA 2016, Part I, LNCS 9827, pp. 287–302, 2016.
DOI: 10.1007/978-3-319-44403-1_18

Similarity between two networks is a function of similarities between their orders, sizes, and topological features. While similarities in orders and sizes are trivial to assess, capturing topological and structural similarities is the core challenge in the task of network[1] comparison. Comparison of two networks essentially entails analogizing structural properties such as nature of hierarchy, clustering tendency, neighborhood characterization, correlation between topological attributes etc. Networks may be compared either at a local or global level depending upon the application. For example, construction of a phylogenetic tree using biological networks involves clustering organisms with similar biological evolution. This task demands a global comparison of networks. On the other hand, comparison of two metabolic networks for the purpose of discovering causal factors for functional differences calls for local level comparison.

Extraction of global features like diameter, average clustering coefficient, characteristic path length, betweenness centrality etc. for comparison purpose is unattractive because of high computational complexity even for medium-sized graphs. Computation of local features, on the other hand, involves examining configuration and properties of small subgraphs, conferring scale independence to the comparison method. Hence, it is tempting to adopt local properties for structural comparison of massive networks. Earlier approaches for network comparison pursued this trend and deployed local features including degree, clustering coefficient, degree centrality, triad census, graphlet distribution etc. [10]. Lamentably, these approaches fail to capture underlying structural hierarchy prevailing in real-life networks.

Therefore, it is desirable to devise methods that summarize network structure both locally and globally. Since hierarchical k-core decomposition promotes the local feature *degree* to the global feature *coreness*, we explore k-core decomposition as a tool to quantify the structural similarity between two networks. K-core decomposition has been recognized as an important technique for understanding complex networks by decomposing them in hierarchy [2,12,24]. We posit that hierarchical segmentation using k-core decomposition method has potential to reveal structural differences between networks at all levels of hierarchy.

1.1 Motivation

Motivational factors for using hierarchical k-core decomposition approach for scalable network comparison are listed below.

i. Real-life networks exhibit structural hierarchy and comparing analogous signals at all levels of hierarchies has potential to reveal the structural disparity between networks.
ii. The proposed algorithm is particularly appealing for comparing large and sparse graph since k-core decomposition method has computational complexity of $O(|\mathcal{E}|)$, $\mathcal{E}$ being the set of edges [4].
iii. Massive networks that cannot fit in main memory can be decomposed using distributed k-core decomposition [22].

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

1.2 Contributions

In this paper, we propose a novel and scalable method for *N*etwork *C*omparison using *k*-core *D*ecomposition (*NCKD*). According to Faust [10], network comparison studies are designed to answer two questions. First, does a pair of networks exhibit common structural tendencies?, and second, which structural features distinguish among different relations between nodes? We demonstrate that node distribution in shells of the network is an effective and efficient implement to answer the first question. Augmenting node distribution in shells with edge distribution boosts its power to cogently answer the second question. Research contributions of the paper are listed below:

i. A novel algorithm (*NCKD*) that uses k-core decomposition to quantify network similarity through network signatures generated using probability distribution of nodes and edges in shells respectively (Sect. 4).
ii. Comparison of *NCKD* with two state-of-the-art network comparison algorithms (Sect. 5.2).
iii. Extensive experimentation to demonstrate effectiveness, scalability and robustness of *NCKD* (Sects. 5.3 and 5.4).

2 Related Work

Several decent algorithms for network comparison, that quantify similarity between networks, have been proposed in recent years. A related but different problem is *network alignment*, addressed in bio-informatics, where the goal is to map nodes of one network to the nodes of another. Our focus is on recent representative network comparison algorithms followed by a brief overview of applications of k-core decomposition.

2.1 Network Comparison

Popular approaches for comparing networks include (i) graph isomorphism, (ii) graph edit distance, (iii) iterative methods, and (iv) feature extraction [6].

Graph isomorphism, a theoretically sound approach, has been traditionally employed to establish exact matching between two graphs [15]. Approximate matching is commonly obtained by graph edit distance, which essentially is an error-tolerant method [11]. Iterative methods compute the pairwise similarity between nodes by capturing similarity/dissimilarity of their neighborhoods [21]. These three approaches lead to algorithms with high computational complexity and are hence non-scalable [18]. This deters their applicability to large networks.

Feature extraction approach has recently found favour with the community interested in analyzing massive graphs. The strategy involves constructing features from the compared graphs and computing distance between them to quantify differences. Banerjee [3] used eigenvalues of normalized graph Laplacian spectra to capture global topological properties for computing pairwise networks similarity. Recently, Lu et al. [18] compared complex networks using the heat content estimated by lazy random walk.

Macindoe et al. [19] considered all induced subgraphs of a parametrized radius centered on each vertex and computed three socially relevant structural features, Leadership, Bonding, and Diversity (L,B,D), driven by social theories for each subgraph. Earth mover's distance between LBD distributions of networks quantifies their similarity. Netsimile [6] algorithm composes network signature from moments of distributions of selected local topological properties of the network. The pairwise similarity score of networks is computed using Canberra distance between their signatures. Scale-independent nature of selected properties renders a computational complexity of $O(N)$, where N is the order of the graph.

These algorithms make use of either local or global features, each of which is individually ineffective and non-scalable for network discrimination. NCKD algorithm plugs the gap as it is scalable and exploits local feature while taking into account the global hierarchical structure of the network.

2.2 K-Core Decomposition

Seidman [24] introduced k-core decomposition for characterizing network structure.The k-core of a network is a maximal subgraph in which every node is connected to at least k other nodes. Batagelj et al. [4] present an $O(|\mathcal{E})|$ algorithm for k-core decomposition of a graph $\mathcal{G}$ with $|\mathcal{E}|$ edges. Analysis of the k-core structure of a graph has been effectively used in identification of social cores and influential nodes in social networks, acceleration of community detection, evaluation of co-operation in communities, and as a visualization tool to highlight the topological and hierarchical structure of graphs [2,12,23]. Recently proposed k-truss decomposition method also presents a hierarchical view of the network yielding the largest subgraph in which every edge is contained in at least (k-2) triangles within it [25]. The method is effective for focusing on smaller and cohesive areas, which are subgraphs of k-core. However higher computational complexity of order $O(|\mathcal{E}|^{1.5})$ for k-truss decomposition is a discouraging factor. Hence, we chose to use k-core decomposition method for network comparison.

To the best of authors' knowledge, NCKD is first-ever application of k-core decomposition for scalable network comparison using single PC.

3 Preliminaries and Notation

We introduce formal notation and definitions used in the paper. Let $\mathcal{G}$ be a simple, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. An edge $\mathsf{e}_{ij} \in \mathcal{E}$ if it connects vertices v_i and v_j; $\mathsf{v}_i, \mathsf{v}_j \in \mathcal{V}$. The order of $\mathcal{G}$ is $|\mathcal{V}|$ and its size is $|\mathcal{E}|$. The degree of a vertex v is denoted by $\rho(\mathsf{v})$. The k-core decomposition algorithm iteratively prunes vertices of degree less than k resulting in a hierarchy of nested k-core sub-graphs, within which each node is connected to at least k other nodes. Formal definitions as adapted from [2] follow.

Definition 3.1. *A subgraph, $\mathcal{G}'_k = (\mathcal{V}'_k, \mathcal{E}'_k)$ of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, induced by the set $\mathcal{V}'_k \subseteq \mathcal{V}$ is a k-core (core of order k) of $\mathcal{G}$ if $\forall \mathsf{v} \in \mathcal{V}'_k$: $\rho(\mathsf{v}) \geq k$, $(k \geq 0)$ and $\mathcal{G}'_k$ is a maximal connected subgraph with this property.* □

Definition 3.2. *Coreness $\zeta(\mathsf{v})$ of vertex v is k if it belongs to a k-core but not to any (k+1)-core. Coreness of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is $max\{\zeta(\mathsf{v}) \,\forall \mathsf{v} \in \mathcal{V}\}$.* □

Definition 3.3. *A k-shell ($\mathcal{S}_k$) of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the set of all vertices with coreness k, i.e., $\mathcal{S}_k = \{\mathsf{v} | \mathsf{v} \in \mathcal{V} \wedge \zeta(\mathsf{v}) = k\}$.* □

The k-core decomposition reflects the structure of a network by faithfully capturing the inherent hierarchy as nested cores. The lower bound on number of nodes in a k-core is (k+1) and a loose upper bound is $|\mathcal{V}|$. The lower bound on the number of edges is $\binom{\mathrm{k}+1}{2}$, while a loose upper bound is $|\mathcal{E}|$ [5]. If both endpoints of an edge have the same coreness, the edge is termed as an *intra-shell* edge, otherwise it is an *inter-shell* edge.

Example 3.1. *Figure 1 shows graph $\mathcal{G}$ with $|\mathcal{V}| = 32$ and $|\mathcal{E}| = 47$. Dashed circles, marked $\mathrm{k} = i$, demarcate the cores. Nodes within a dashed circle and having same color denote shell $\mathcal{S}_k$. Shell $\mathcal{S}_4$, the highest order shell induces the 4-core of $\mathcal{G}$. Subgraph induced by $\mathcal{S}_3 \cup \mathcal{S}_4$ is the 3-core of $\mathcal{G}$.* □

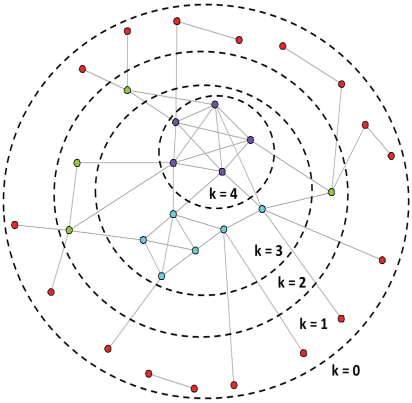


Fig. 1. K-core decomposition of $\mathcal{G}$. Nodes with same color constitute a shell. (Color figure online)

4 Characterizing Networks Using K-Core Decomposition

Adaptation of k-core decomposition for designing a similarity measure is non-trivial because two networks with the same hierarchical structure can have vastly different topology. The challenge is to identify and extract suitable features of the decomposed graph for effective and scalable network discrimination. We hypothesize that differences in the node/edge distribution of shells in the decomposed graph are effective discriminators for the overall structure of underlying networks. We first explain a simple and effective network feature i.e. node distribution followed by the statement of its limitation, and reasoning behind inclusion of edges arrangements.

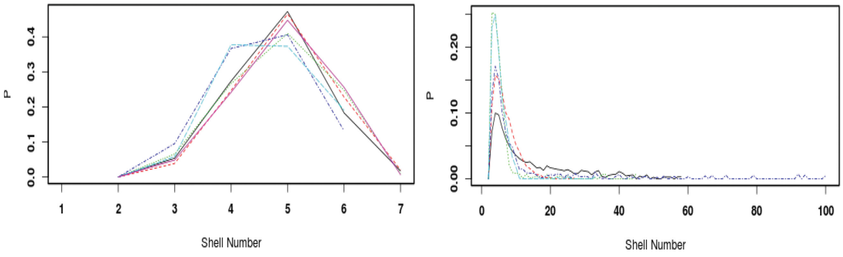
4.1 Coreness Distribution

Distribution of nodes within shells captures the spread of nodes and reflects the underlying structure [24]. It is synonymous with the distribution of coreness of nodes in the decomposed graph.

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with coreness k and shells $\{\mathcal{S}_0, \ldots, \mathcal{S}_\mathsf{k}\}$. Let X be a discrete random variable denoting coreness of a node in $\mathcal{G}$ and defined on the sample space $\Lambda = \{0, \ldots, \mathsf{k}\}$. We define probability mass function for X as $p(x) = p(X = x) = \frac{|\mathcal{S}_x|}{|\mathcal{V}|}$, where $|\mathcal{S}_x|$ is the cardinality of shell $\mathcal{S}_x$. It is clear that $\sum_{x=0}^{\mathsf{k}} |\mathcal{S}_x| = |\mathcal{V}|$. Here, $p(x)$ denotes the probability that a node has coreness value x. Alternatively, $p(x)$ is the probability of an arbitrary node lying in shell $\mathcal{S}_x$. Following example explains computation of probability distribution (p) of nodes (coreness) in the shell.

Example 4.1. *Graph $\mathcal{G}$ in Fig. 1 has 32 nodes and 5 shells. Probability distribution (p) of nodes in $\mathcal{G}$, is given by $p = \langle 0/32,\ 17/32,\ 4/32,\ 6/32,\ 5/32 \rangle$.*

We studied probability distribution of coreness for several synthetic and real-life networks (Table 1) to test its propensity for network comparison. We show the plot of coreness distribution of six metabolic and five co-author networks in Figs. 2a and b. The striking similarity between the coreness probability distributions of graphs belonging to the same genre strongly indicates its utility as a discriminating network feature. Preliminary experimentation (not reported due to space constraint) however quickly revealed the inadequacy of this feature to effectively capture structural differences arising in the real world networks.



(a) Metabolic Networks (b) Co-authors Networks

Fig. 2. Plots for probability distribution of coreness for two genres of real-life networks.

This insufficiency arises because the arrangement of edges in a graph, which is the cause of topological variations, is completely ignored by the coreness distribution. Extreme topologies of star and chain with n nodes having identical hierarchical structure and probability distribution of coreness, present a very clear example to substantiate the argument. Since *coreness is inadequate to capture finer structural differences between networks, it is myopic to depend on it as a sole distinguishing feature.*

4.2 Edges Distribution

Theoretically, a graph of order n with coreness k and coreness distribution p is a random sample from the family $\mathcal{F}_{\mathsf{k},p,n}$ of graphs [14]. Graph $\mathcal{G}$ in question is one realization from this family. All graphs in $\mathcal{F}_{\mathsf{k},p,n}$ will have similar coreness distribution, even though they may be topologically different. This is unacceptable in both theory and practice. *Rewiring* and *swapping* lemmas stated in [5] reinforce this argument.

According to the *rewiring* lemma, two adjacent nodes in a shell can disconnect and connect independently to nodes with higher coreness, and vice versa without changing the coreness distribution of the graph. The *swapping* lemma allows non-adjacent nodes in the same shell to swap end-nodes without altering the coreness distribution of the graph. It is reasonable to conclude that *coreness distribution is inadequate to capture finer structural differences between networks.* For better discrimination between the members of $\mathcal{F}_{\mathsf{k},p,n}$ family, we incorporate arrangements of edges influencing the network structure in addition to nodes distribution.

Let $\mathcal{G}$ be a graph with coreness k. Then, E^l denotes the lower triangular matrix representing the arrangement of edges of G. E^l_{ij} is the count of inter-shell links between shells $\mathcal{S}_i$ and $\mathcal{S}_j$. E^l_{ii} is the count of intra-shell links in shell $\mathcal{S}_i$. Clearly $\sum_i \sum_j E^l_{ij} = |\mathcal{E}|$, $(0 \leq i \leq \mathsf{k}, 0 \leq j \leq i)$. Example 4.2 clarifies the idea of intra- and inter-shells links using matrix representation used in [5].

Example 4.2. *The count of intra- and inter-shell edges in $\mathcal{G}$ of Fig. 1 is shown below in the lower triangular matrix (E^l). Shell $\mathcal{S}_2$ has one intra-shell link indicated by $E^l_{22} = 1$. It also has six inter-shell links with $\mathcal{S}_1$ indicated by $E^l_{21} = 6$.*

$$E^l = \begin{pmatrix} 0 & & & & \\ 0 & 4 & & & \\ 0 & 6 & 1 & & \\ 0 & 5 & 2 & 9 & \\ 0 & 1 & 5 & 4 & 10 \end{pmatrix}$$

We vectorize matrix E^l to a vector V of size $\frac{(\mathsf{k}+1)(\mathsf{k}+2)}{2}$ such that $V = [E^l_{00}, E^l_{10}, E^l_{11}, E^l_{20}, \ldots, E^l_{\mathsf{kk}}]$. Index r in V for E^l_{ij} is obtained by using the following rule.

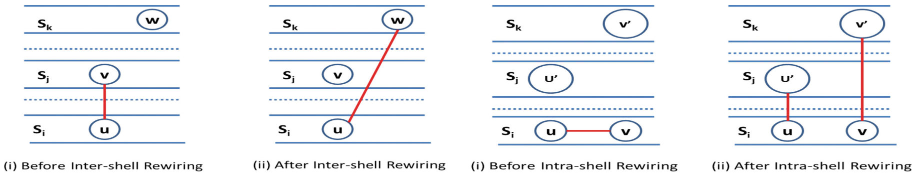
$$r \longleftarrow j + \frac{i * (i+1)}{2} \tag{1}$$

Clearly, the vectorization expresses isomorphism between V and E^l. Let R be a discrete random variable defined on sample space $\Lambda' = \left(0, 1, \ldots, \frac{(\mathsf{k}+1)(\mathsf{k}+2)}{2} - 1\right)$ denoting linkage count within and between shells in the graph. When $R = r$, it denotes linkage between shells $\mathcal{S}_i$ and $\mathcal{S}_j$, with the mapping defined by Eq. 1.

The probability mass function $u(r)$ of random variable R is defined as $u(r) = p(R = r) = \frac{E^l_{ij}}{|\mathcal{E}|}$. Here $u(r)$ denotes the probability that an arbitrary edge in $\mathcal{G}$ connects a node in $\mathcal{S}_i$ to a node in $\mathcal{S}_j$. It is easy to show that u corresponds to probability distribution of intra-shell and inter-shell links. Following example shows the edge probability distribution u for the lower triangular matrix given in Example 4.2.

Example 4.3. *Given 47 edges in $\mathcal{G}$, probability distribution of edges (u) is computed as: ⟨ 0/47, 0/47, 4/47, 0/47, 6/47, 1/47, 0/47, 5/47, 2/47, 9/47, 0/47, 1/47, 5/47, 4/47, 10/47 ⟩.* □

If the number of nodes and edges in two graphs with identical coreness distribution are same, structural differences between them arise due to rewiring of edges. To examine the sensitivity of E^l towards rewiring, we focus on the cases that do not alter the coreness of the involved nodes post-rewiring. There are two possibilities for rewiring of a node. It can either connect to a node in the same shell or to a node in a higher shell. Rewiring of a node in a lower shell can be considered as the former situation from the viewpoint of the node in the lower shell. We explain these cases below.


Sk
w
Sj
v
Si
u
(i) Before Inter-shell Rewiring
Sk
w
Sj
v
Si
u
(ii) After Inter-shell Rewiring
Sk
v'
Sj
u'
Si
u
v
(i) Before Intra-shell Rewiring
Sk
v'
Sj
u'
Si
u
v
(ii) After Intra-shell Rewiring

(a) *Inter-shell* (b) *Intra-shell*

Fig. 3. Example for edge rewiring. $\mathcal{S}_i, \mathcal{S}_j, \mathcal{S}_k$ are the shells.

R1. *Rewiring an inter-shell edge*: Consider nodes $\mathsf{u}, \mathsf{v}, \mathsf{w} \in \mathcal{V}$, located in distinct shells $\mathcal{S}_i$, $\mathcal{S}_j$, $\mathcal{S}_k$ respectively, s.t. $i \neq j \neq k$ and $i < j, k$. Let edges $(\mathsf{u}, \mathsf{v}) \in \mathcal{E}$ and $(\mathsf{u}, \mathsf{w}) \notin \mathcal{E}$ Then R1 leads to

$$\mathcal{E} := \mathcal{E} \setminus (\mathsf{u}, \mathsf{v}) \cup (\mathsf{u}, \mathsf{w}) \tag{2}$$

Figure 3(a) exhibits this case. Since node u lies in shell $\mathcal{S}_i$, it has at least i links with nodes in higher shells. After deleting edge (u, v) and adding edge (u, w), link count of u in higher shell remains same. Consequently, coreness of u remains unchanged. Coreness of v and w remains unchanged since links to lower shells do not impact coreness (by Definition 3.1). Consequently, post-rewiring coreness distribution remains unchanged. In the edge distribution matrix, two entries change as follows: E^l_{ik} is incremented and E^l_{ij} is decremented by 1. Hence altered structure of the graph is captured by the edge distribution.

R2. *Rewiring an intra-shell edge*: Consider nodes $\mathsf{u}, \mathsf{v}, \mathsf{u}', \mathsf{v}' \in \mathcal{V}$ s.t. $\mathsf{u}, \mathsf{v} \in \mathcal{S}_i$, $\mathsf{u}' \in \mathcal{S}_j$, $\mathsf{v}' \in \mathcal{S}_k$ and $i \neq j \neq k$, $i < j, k$. Let edges $(\mathsf{u}, \mathsf{v}) \in \mathcal{E}$, $(\mathsf{u}, \mathsf{u}') \notin \mathcal{E}$, $(\mathsf{v}, \mathsf{v}') \notin \mathcal{E}$. Then R2 leads to

$$\mathcal{E} := \mathcal{E} \setminus (\mathsf{u}, \mathsf{v}) \cup (\mathsf{u}, \mathsf{u}') \cup (\mathsf{v}, \mathsf{v}') \tag{3}$$

Figure 3(b) exhibits this case. Following similar arguments as in R1, intra-shell rewiring does not alter coreness distribution, but is reflected in edge distribution.

4.3 NCKD Algorithm

The proposed algorithm for *N*etwork *C*omparison using *k-core D*ecomposition (*NCKD*) uses probability distribution of *nodes* as well as intra-shell/inter-shell *edges*. The problem of pair-wise network comparison reduces to computing the statistical distance between probability distributions representing signatures of the networks. We use *Jensen-Shannon Distance* (JSD) as it is a popular metric for comparing probability distributions due to its property of non-negativity, identity, symmetry, and boundedness [17]. Equation 4 gives JSD between two probability distributions p and q, with respective weights w_1 and w_2 ($w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$).

$$JSD(p, q) = [H(w_1 * p + w_2 * q) - w_1 * H(p) - w_2 * H(q)]^{\frac{1}{2}} \tag{4}$$

Here, H is the Shannon entropy function. Equipped with a tool to capture finer distinctions of graph topologies, we quantify the structural difference (distance) between two networks as average of differences (distance) between the (i) distribution of coreness and (ii) distribution of edges.

Let p and q respectively denote the probability distributions of coreness of graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. Further, u and v denote the edge probability distributions of graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. Applying JSD on these distributions and averaging the result gives the net distance between two networks. Equation 5 formally defines distance between networks.

$$\mathcal{D}(\mathcal{G}_1, \mathcal{G}_2) = avg(JSD(p, q), JSD(u, v)) \tag{5}$$

Algorithm 1. Algorithm *NCKD*

```
Input  : Graphs $\mathcal{G}_1$and $\mathcal{G}_2$
Output: Distance between $\mathcal{G}_1$and $\mathcal{G}_2$
Begin
  Decompose $\mathcal{G}_1$and $\mathcal{G}_2$ into cores
  $p \leftarrow$ Prob. dist. of coreness of nodes in $\mathcal{G}_1$
  $q \leftarrow$ Prob. dist. of coreness of nodes in $\mathcal{G}_2$
  $u \leftarrow$ Prob. dist. of intra and inter-shell links in $\mathcal{G}_1$
  $v \leftarrow$ Prob. dist. of intra and inter-shell links in $\mathcal{G}_2$
  $\mathcal{D}(\mathcal{G}_1, \mathcal{G}_2) \leftarrow avg(JSD(p, q), JSD(u, v))$ //Jensen-Shannon Distance
End
```

Algorithm 1 summarizes the steps for *NCKD*. Please note that in all experiments reported in paper, we assign equal weights to the distributions while computing JSD. We are conscious that weights can be constructively manipulated to capture preference for one graph over other during the comparison.

5 Experiments

The experimental study is designed to assess and compare effectiveness, scalability and robustness of *NCKD* algorithm against two recent network comparison algorithms *Netsimile* [6] and *LBD* [19]. We implemented *NCKD* algorithm in *Python* (64 bits, v 2.7.3) and executed on Intel Core i5-3201M CPU @2.50 GHz with 8 GB RAM, running UBUNTU 12.04.

5.1 About Datasets

We performed experiments with both synthetic and real-life datasets (Table 1). Synthetic datasets allow controlled variation of data characteristics and hence enable close scrutiny of algorithmic behaviour. Real-life datasets expose the strengths and weakness of the algorithm in practical scenarios.

Synthetic datasets were generated using *igraph* package of *R*. Erdös-Rényi (E), Forest-Fire (F), Watts-Strogatz (W), and Barabási-Albert (B) models were used for analysis. Order (number of nodes) of the network in thousands (K) is included in nomenclature. Since each network is one probabilistic realization of the model parameters, we generated multiple networks with same parameters. Thus, B10K-n meant n^{th} realization of Barabási-Albert network of order 10K. Three real-life genres include (i) Co-author (CA), (ii) Autonomous Systems (A), and (iii) Metabolic (M) networks. Large datasets used for scalability experiment are described in Sect. 5.4.

5.2 Effectiveness of *NCKD*

We compute effectiveness of *NCKD* by comparing $\mathcal{D}(\mathcal{G}_1, \mathcal{G}_2)$ (Eq. 5) with distance measures defined in two state-of-the-art algorithms *LBD* and *Netsimile*. We compute pairwise distances for networks given in Table 1, using distance measures used in three algorithms. *LBD* algorithm was unable to generate network signatures for large graphs even after running for more than 24 h. We, therefore, restrict experiment to 14/32 graphs that *LBD* algorithm was able to process in reasonable time (<4 h) and cluster the networks using hierarchical agglomerative clustering[2]. We compute purity, precision, recall, accuracy, and Normalized Mutual Information (NMI) measures [20] to assess the quality of clustering (Table 2).

It is clear from Table 2 that resultant clustering of 14 small networks by *NCKD* is better than those delivered by *Netsimile* and *LBD* algorithms. Timings (averaged over 3 runs) for generating network signatures (Table 3) for *small* networks show that *NCKD* is also faster. We dropped *LBD* algorithm for further experimentation since it was patently non-scalable and the clustering, as evidenced by NMI, was also poorer in quality.

[2] *hclust* and *cutree* functions of *stats* package in *R* were used for agglomerative clustering and to cut dendrogram by specifying known number of classes.

Table 1. Characteristics of synthetic and real-life networks used; D: Diameter, C: Connected components, GCC: Global clustering coefficient, α: Parameter for power law distribution

Networks		Nodes	Edges	D	C	GCC	α	Remarks
Synthetic networks using generative models								
Erdos Reyni [8]	E10K-1	9827	19823	15	7	0.0004	11.47	Generator G(n,m = 2n)
	E10K-2	9807	19772	14	9	0.0005	11.58	m: number of edges
	E1K-1	983	2010	10	3	0.0030	13.568	n: number of nodes
	E1K-2	984	2022	10	3	0.0037	10.61	
Forest Fire [16]	F10K-1	10000	58901	6	1	0.0598	3.06	4 ambassador vertices
	F10K-2	10000	58823	6	1	0.0588	3.105	20 % backward burning probability
	F1K-1	1000	5873	5	1	0.0894	3.05	30 % forward burning probability
	F1K-2	1000	5717	5	1	0.0899	3.111	
Watts Strogatz [26]	W20K-1	20000	80000	8	1	0.0714	8.30	Lattice dimension = 1
	W20K-2	20000	80000	8	1	0.0694	8.25	Degree = 4
	W2K-1	2000	8000	7	1	0.0757	8.19	Rewiring probability = 0.3
	W2K-2	2000	8000	7	1	0.0731	8.22	
Barabasi Albert [1]	B10K-1	10000	9999	2	1	0	1.33	Non-assortative version
	B10K-2	10000	9999	3	1	0	1.33	added 4 edges/iteration
	B1K-1	1000	999	3	1	0	2	
	B1K-2	1000	999	2	1	0	2	
Real-life networks								
Co-author [16]	CA-1	18772	396159	14	290	0.3180	1.71	Papers submitted to arXiv during
	CA-2	23133	186935	15	567	0.2643	2.21	period January 1993 to April 2003
	CA-3	5242	28979	17	355	0.6298	2.23	Astro Physics, Condensed Matter
	CA-4	12008	237009	13	278	0.6595	1.74	General Relativity, High Energy
	CA-5	9877	51970	18	429	0.2840	2.36	Physics (HEP) and HEP Theory
Autonomous [16]	A-1	10670	22002	10	1	0.0093	2.17	Oregon route-views for period
	A-2	10729	21999	12	1	0.0085	2.19	March 31 to May 26, 2001
	A-3	10790	22469	10	1	0.0094	2.20	
	A-4	10859	22747	10	1	0.0097	2.206	
	A-5	10886	22493	10	1	0.0089	2.19	
Metabolic [13]	M-1	1268	3011	14	1	0	2.17	Three types of Organisms
	M-2	490	1163	11	1	0	2.18	Archaea (M-1, M-2),
	M-3	993	2368	12	2	0	2.21	Bacteria (M-3, M-4)
	M-4	409	880	9	7	0	2.35	and Eukaryotes (M-5, M-6)
	M-5	665	1514	14	3	0	2.25	
	M-6	1511	3833	14	1	0	2.37	

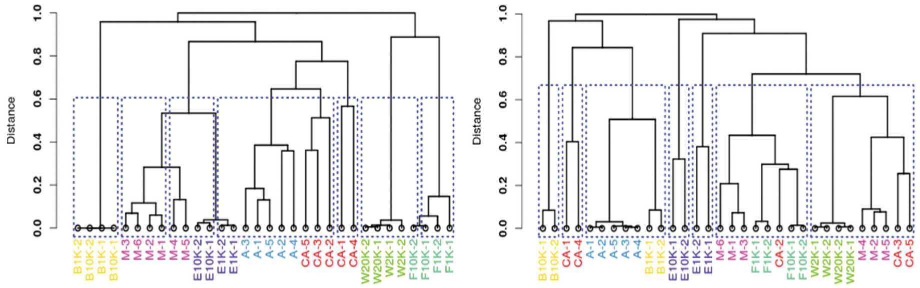
Next, we execute *Netsimile* and *NCKD* on all networks mentioned in Table 1. Figures 4a and b show the dendrograms generated from pairwise distances computed by two algorithms. It is evident that *NCKD* algorithm performs better grouping than *Netsimile*. Cluster quality metrics of 32 networks (Table 2) for two algorithms vindicate the visual observation. Comparison of execution time of two algorithms reveals that *NCKD* is several orders faster than *Netsimile* (See *Large* synthetic and real-life networks in Table 3). The swift execution of *NCKD* indicates its scalability.

Table 2. Quality metrics for hierarchical clustering of 14 small networks and all 32 networks.

Datasets	Algorithm	Purity	Precision	Recall	Accuracy	NMI
Small networks (14)	*NCKD*	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	LBD	0.8571	0.8182	0.9474	0.9451	0.8921
	NetSimile	0.8947	0.625	0.5263	0.8352	0.8213
All networks (32)	*NCKD*	**0.875**	**0.688**	**0.8983**	**0.9395**	**0.9161**
	Netsimile	0.656	0.382	0.5763	0.8387	0.6885

Table 3. Signature generation time (in seconds) for *NCKD*, *Netsimile* and *LBD* on selected networks from Table 1. A - indicates that the algorithm did not complete even after running for 24 h.

Algorithm → Networks ↓		*NCKD*	*Netsimile*	*LBD*
Small	M-2	**0.015**	0.783	120.615
	F1K-1	**0.018**	0.899	1825.11
	CA-5	**0.076**	6.711	6583.153
Large synthetic	B10K-1	**0.0202**	111.534	–
	E10K-1	**0.0503**	4.817	–
	W20K-1	**0.097**	25.353	–
	F10K-1	**0.077**	13.643	–
Large real-life	M-6	**0.013**	0.821	–
	A-5	**0.065**	30.676	–
	CA-1	**0.245**	198.789	–

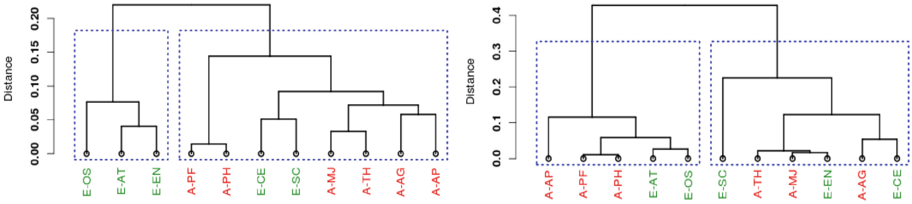


(a) *NCKD*

(b) *Netsimile*

Fig. 4. Dendrogram for networks described in Table 4. Networks belonging to same genre have same color. (Color figure online)

In order to capture finer distinctions between networks of the same genre, we selected eleven metabolic networks in two sub-categories (six Archaea (A) and five Eukaryotes (E)) whose order ranges from 490 to 1511 and size ranges from 1148 to 3807 [13]. We performed hierarchical agglomerative clustering of these networks from distance matrices generated by *NCKD* and *NetSimile* (Fig. 5). Algorithm *NCKD* is able to identify one pure group of Eukaryotes, which *Netsimile* missed. The clustering quality metrics for metabolic networks shown in Table 4 confirm the effectiveness of *NCKD* over *Netsimile*.



(a) *NCKD*

(b) *Netsimile*

Fig. 5. Metabolic networks in two sub-categories - A: Archaea, and E: Eukaryote.

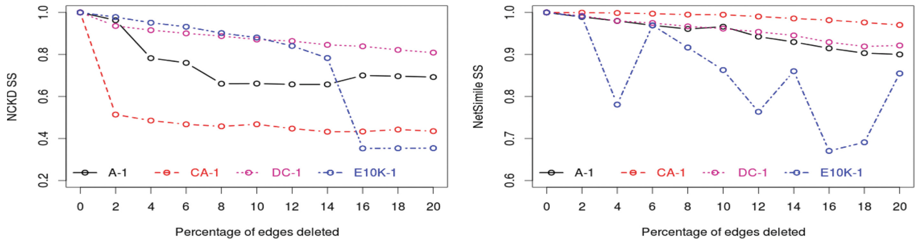
Table 4. Quality metrics for dendrograms shown in Fig. 5

N/w type →	Metabolic networks				
Algorithm ↓	Purity	Precision	Recall	Accuracy	NMI
NCKD	**0.8182**	**0.6129**	**0.76**	**0.6727**	**0.4393**
Netsimile	0.5455	0.40	0.40	0.4545	0.0073

5.3 Handling of Missing Data

We compare effectiveness of *NCKD* and *Netsimile* towards missing data. For this experiment, we compared networks with themselves after applying random edge deletion systematically. For network G, we created $G'_{x_1}, G'_{x_2} \cdots G'_{x_k}$ variations by deleting x_i% of edges from it. Intuitively, both algorithms should yield similarity score (SS) of 1 while comparing G with G'_0, with the score falling as deleted edges increase. Fall in SS is expected to be different for different graphs due to structural differences. In order to beat the effect of randomness, reported results are averaged over three runs.

Three real-life networks (A-1, DC-1, CA-1) and one synthetic network (E10K-1) were perturbed by deleting edges from 0 % to 20 % (in steps of 2) and compared using two algorithms. Similarity scores obtained by *NCKD* and *Netsimile* are plotted in Figs. 6a and b. *Netsimile* registered a fall of maximum 10 % for the real-life datasets even after deleting 20 % edges while *NCKD* revealed significant differences in networks. This observation indicates superior ability of *NCKD* to suitably react to missing data.



(a) Variation in SS - *NCKD*

(b) Variation in SS - *Netsimile*

Fig. 6. Comparison of robustness towards missing data - *NCKD* vs. *Netsimile*.

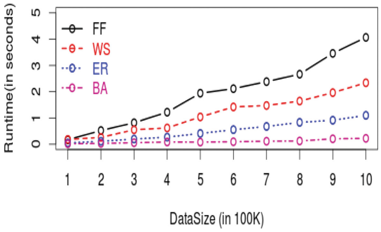


Fig. 7. Feature generation time of *NCKD* for synthetic networks

Large networks	**Nodes** (10^6)	**Edges** (10^6)	**Runtime (seconds)**
Amazon product	0.33	0.93	0.799
Road n/w of Texas	1.38	1.92	1.528
Road n/w of California	1.96	2.77	3.4824
Youtube OSN	1.13	2.99	3.532
Web graph of Berkeley and Stanford	0.69	7.796	10.889

Fig. 8. Feature generation time of *NCKD* for *massive* real-life networks. n/w: Network

5.4 Scalability w.r.t Large Datasets

Networks generated from different models allow convenient variations in the order of graphs to examine scalability of *NCKD*. We generated 10 graphs for each generative model (description in Table 1) with varying number of nodes 100K to 1000K in steps of 100K, and edges proportionally depending on the model. *Netsimile* was unable to process graphs of order >100K even after running for more than 24 h. Hence, it was dropped for scalability analysis. *NCKD* was executed five times for each graph to average out the timing observations. Figure 7 shows approximately sublinear growth in timings for each model. The increase in timings for the models varies with the number of edges in the corresponding networks. Edges increase fastest in FF model and slowest in BA model, which is faithfully reflected by the timings for two models.

Figure 8 shows execution timings of *NCKD* for five real-life large datasets downloaded from SNAP[3], which strengthens the claim of scalability. Since k-core decomposition algorithm is $O(\mathcal{E})$, time increases linearly with edges.

6 Conclusion

Each large-scale network is unique at the microscopic level. However, at different levels of resolutions, commonalities emerge among different pairs of graphs.

[3] http://snap.stanford.edu/data.

Discovery of these commonalities and their quantification is the goal of the proposed algorithm *NCKD* (Network Comparison using k-core Decomposition), which is intuitive, effective and scalable. The algorithm decomposes the graph into cores, analyses shells and constructs node and edge related probability distributions, which serve as network signatures. Jensen-Shannon distance is applied on these signatures to find distance between networks. We establish that node and edge distributions adequately discriminate networks.

Extensive experimentation and comparison of *NCKD* with *Netsimile* and *LBD* algorithms establish its superiority in terms of effectiveness and scalability. Execution timings for large synthetic and real-life networks affirm its scalability. We also demonstrate that *NCKD* is sensitive to the underlying topological structure of the graph, but needs to be improved to take cognizance of size and order of the network. The agenda for future is to overcome its deficiency to clearly segregate networks of the same genre by including more shell features.

References

1. Albert, R., lászló Barabsi, A.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47 (2002)
2. Alvarez-Hamelin, J.I., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. Adv. Neural Inf. Process. Syst. **18**, 41–50 (2006)
3. Banerjee, A.: Structural distance and evolutionary relationship of networks. Biosystems **107**(3), 186–196 (2012)
4. Batagelj, V., Zaversnik, M.: An O(m) algorithm for cores decomposition of networks. CoRR cs.DS/0310049 (2003)
5. Baur, M., Gaertler, M., Grke, R., Krug, M.: Generating graphs with predefined k-core structure. Technical report, DELIS - Dynamically Evolving, Large-Scale Information Systems (2007)
6. Berlingerio, M., Koutra, D., Eliassi-Rad, T., Faloutsos, C.: Network similarity via multiple social theories. In: Proceedings of International Conference on ASONAM, pp. 1439–1440. IEEE (2013)
7. Dorogovtsev, S., Goltsev, A., Mendes, J.: Critical phenomena in complex networks. Rev. Mod. Phys. **80**, 1275 (2008)
8. Erdös, P., Rényi, A.: On random graphs I. Publicationes Math. **6**, 290–297 (1959). Debrecen
9. Faloutsos, C., Koutra, D., Vogelstein, J.T.: DELTACON: a principled massive-graph similarity function. In: Proceedings of the 13th SIAM International Conference on Data Mining, pp. 162–170 (2013)
10. Faust, K.: Comparing social networks: size, density and local structure. Adv. Methodol. Stat. **3**(2), 185–216 (2006)
11. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. **13**(1), 113–129 (2010)
12. Giatsidis, C., Thilikos, D.M., Vazirgiannis, M.: Evaluating cooperation in communities with the k-core structure. In: Proceedings of International Conference on ASONAM, pp. 87–93. IEEE (2011)
13. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabasi, A.L.: The large-scale organization of metabolic networks. Nature **407**(6804), 651–654 (2000)

14. Karwa, V., Pelsmajer, M.J., Petrovic, S., Stasi, D., Wilburne, D.: Statistical models for cores decomposition of an undirected random graph. CoRR abs/1410.7357 (2014)
15. Kollias, G., Mohammadi, S., Grama, A.: Network similarity decomposition (NSD): a fast and scalable approach to network alignment. Technical report. Purdue University (2011)
16. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. ACM Trans. Knowl. Discov. Data **1**(1), 2 (2007)
17. Lin, J.: Divergence measures based on the shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
18. Lu, S., Kang, J., Gong, W., Towsley, D.: Complex network comparison using random walks. In: Proceedings of 23rd International WWW Conference, pp. 727–730 (2014)
19. Macindoe, O., Richards, W.: Graph comparison using fine structure analysis. In: Proceedings of the 2nd IEEE International Conference on Social Computing, pp. 193–200 (2010)
20. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
21. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In: Proceedings of the 18th ICDE, pp. 117–128 (2002)
22. Montresor, A., Pellegrini, F.D., Miorandi, D.: Distributed k-core decomposition. IEEE Trans. Parallel Distrib. Syst. **24**(2), 288–300 (2013)
23. Peng, C., Kolda, T.G., Pinar, A.: Accelerating community detection by using k-core subgraphs. CoRR abs/1403.2226 (2014)
24. Seidman, S.B.: Network structure and minimum degree. Soc. Netw. **5**, 269–287 (1983)
25. Wang, J., Cheng, J.: Truss decomposition in massive networks. Proc. VLDB Endowment **5**(9), 812–823 (2012)
26. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. Nature **393**(6684), 440–442 (1998)