Physica A 497 (2018) 166–184

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Compressive sensing of high betweenness centrality nodes in networks

Hamidreza Mahyar [a], Rouzbeh Hasheminezhad [b], Elahe Ghalebi K. [c], Ali Nazemian [d], Radu Grosu [c], Ali Movaghar [d], Hamid R. Rabiee [d,*]

[a] Faculty of Science and Engineering, International Campus-Kish Island, Sharif University of Technology (SUT), Iran
[b] Department of Computer Science, ETH Zurich (ETHZ), Switzerland
[c] Institute of Computer Engineering, Vienna University of Technology (TU Wien), Austria
[d] Department of Computer Engineering, Sharif University of Technology (SUT), Iran

HIGHLIGHTS

- CS-HiBet is a new approach to detect the top-k betweenness centralities in networks.
- Our method uses compressive sensing via indirect end2end (aggregated) measurements.
- CS-HiBet can perform as a distributed algorithm using only local node information.
- The proposed method is applicable to large-scale real-world and synthetic networks.
- The experiments show the superiority of CS-HiBet compared to the existing methods.

ARTICLE INFO

Article history:
Received 30 June 2017
Received in revised form 18 November 2017
Available online 16 January 2018

Keywords:
Compressive sensing
Betweenness centrality
Complex network

ABSTRACT

Betweenness centrality is a prominent centrality measure expressing importance of a node within a network, in terms of the fraction of shortest paths passing through that node. Nodes with high betweenness centrality have significant impacts on the spread of influence and idea in social networks, the user activity in mobile phone networks, the contagion process in biological networks, and the bottlenecks in communication networks. Thus, identifying k-highest betweenness centrality nodes in networks will be of great interest in many applications. In this paper, we introduce CS-HiBet, a new method to efficiently detect top-k betweenness centrality nodes in networks, using compressive sensing. CS-HiBet can perform as a distributed algorithm by using only the local information at each node. Hence, it is applicable to large real-world and unknown networks in which the global approaches are usually unrealizable. The performance of the proposed method is evaluated by extensive simulations on several synthetic and real-world networks. The experimental results demonstrate that CS-HiBet outperforms the best existing methods with notable improvements.

1. Introduction

A wide range of real-world systems can be structured and modeled by the means of networks, where actors of the system are indicated by nodes (vertices), and the existing connections between nodes are demonstrated by links (edges).

* Corresponding author.
E-mail addresses: hmahyar@ce.sharif.edu (H. Mahyar), shashemi@student.ethz.ch (R. Hasheminezhad), eghalebi@cps.tuwien.ac.at (E. Ghalebi K.), nazemian@ce.sharif.edu (A. Nazemian), radu.grosu@tuwien.ac.at (R. Grosu), movaghar@sharif.edu (A. Movaghar), rabiee@sharif.edu (H.R. Rabiee).

https://doi.org/10.1016/j.physa.2017.12.145
0378-4371/

The network links can be undirected or directed, and weighted or unweighted due to the nature of nodes interactions. The well-known examples of such networks include technological and transportation infrastructures, communication systems, biological systems, information systems, and a variety of social interaction structures [1,2]. Identifying important nodes has been a fundamental problem in structural analysis of networks [3–8]. The concept of network centrality, a key feature in social network analysis, measures the relative importance of nodes in a network based on their prominence in the graph structure. Various centrality indices have been developed as functions to quantitatively evaluate a node's importance from different points of view [9]. Some of them, *e.g.* degree centrality, give consideration to local properties of the underlying network, while others, *e.g.* betweenness centrality, reflect information about the global network structure [10]. In this paper, we are interested in betweenness centrality [6] which quantifies the fraction of all shortest paths from the nodes to each other that pass through a certain node. Detection of high betweenness centrality nodes is relevant to problems such as identifying important nodes that control flows of information between separate communities of a network [11,12], and detection of causal nodes that highly influence other nodes' behavior (*e.g.* genes in genomics or customers in marketing studies [4,13]). Betweenness centrality has been widely used in analyzing social networks [14–17] and protein networks [18], identifying significant nodes in wireless ad-hoc networks [19], investigating activity of nodes in mobile phone call networks [20], extracting interaction patterns of players on online games [21], identifying key bloggers in dynamic networks of blog posts [22,23], measuring network traffic flow in communication networks [24,25], and also improving recommendations list in social recommender systems [26].

Motivated by numerous applications in the literature, various exact and approximation algorithms have been developed to calculate the betweenness centrality of nodes in the networks [27–32]. The fastest known exact algorithm to compute betweenness centrality of nodes in a network is due to Brandes [28] that requires $O(n + |E|)$ space and a running time of $O(n|E|)$ on unweighted and $O(n|E| + n^2 \log(n))$ on weighted networks, where $|E|$ is the number of links and n is the number of nodes in the network. This algorithm is still prohibitively expensive and impractical for today's networks. Meanwhile, some randomized and approximation methods have been proposed with acceptable theoretical or experimental guarantees in some cases [27,29–31]. One of the main drawbacks of these approaches is that they assume full knowledge of the *network topological structure* which is often unrealistic because networks owners are unwilling to share their topological structure (*i.e.* who is connected to whom) due to privacy concerns and regulations [33,34]. Complete structure for many real networks is initially unavailable because there are access limitations such as login requirements, topological constraints, API query limits, and treatment of user data as proprietary. Thus, in the analysis of many networks, existence of missing data is almost inevitable because the aforementioned constraints may prevent access to entire data of the networks [31,35,36].

In many applications we are only interested in detecting the k-highest betweenness centrality nodes in the network. This is reasonable since a node with a higher centrality is viewed as a more important node than a node with a lower centrality, and in many applications we are only interested in the most important nodes, such as finding influencers in a social network [37,38], and locating bottlenecked junctions/routers in a transportation network/the Internet [25]. Moreover, a community detection application, one of the most well-known applications of the betweenness centrality, utilizes only the highest centrality nodes [11]. Thus, it is often crucial to efficiently and accurately identify the top-k betweenness centrality nodes of a network, while the exact order in top-k list as well as the exact value of the node centrality are not so important [39,40]. To this end, when the global topological structure of a network is known, the straightforward solution to obtain the k-highest betweenness centrality nodes would be to compute the betweenness measure for all the nodes using one of the above randomized or approximation algorithms, then utilize one of the standard sorting algorithms such as Quick sort. However, even the modest average complexity of these algorithms can be still very high for large real networks. Thus, the need for developing algorithms to efficiently identify central nodes in real-world networks seems essential.

To address the aforementioned challenges, *sampling based approaches* have been widely investigated in recent years [41,42,29,22,43–46]. These methods must perform two consecutive steps for estimating characteristics of a network [46]: First, a subset of nodes in the network must be sampled; Second, characteristics of interesting nodes must be estimated in the induced sub-graph of sampled nodes. For estimating the k-highest betweenness centrality nodes, these two steps yield two sources of error. The first one is due to the fact that only a partial view of the network is available through sampling which is referred to as *sampling (collection) error*. The second one is based on the fact that even if a complete view of the induced sub-graph is available, identification of the top-k betweenness centrality nodes might not be totally accurate which is referred to as *identification error*. Beside these errors, the major drawback of these algorithms is that they assume *direct measurement* of each individual node in the network, which can be operationally difficult, costly and sometimes impossible, because of massive scale, distributed management and access limitations of real-world networks. To this end, the topic of inferring network internal features from indirect end-to-end (aggregated) measurements becomes remarkable. Consequently, proposing a new approach to efficiently detect the k-highest betweenness centrality nodes in a network with *indirect measurements* and without full knowledge of the network topological structure is an inevitable task in analysis of real networks. In this paper, we propose a new technique to address this problem and to overcome the aforementioned shortcomings.

The rest of the paper is organized as follows: In Section 2, we explain the limitations of the previous work and the main contributions of the proposed method. We discuss the preliminaries in Section 3. Section 4 provides details of the proposed method. Complexity analysis of our approach is presented in Section 5. Datasets and experimental settings for the evaluation of our method are in Section 6.1 and Section 6.3, respectively. We then provide detailed results of our evaluation in the rest of Section 6. Finally, we conclude the paper in Section 7.

2. Limitations of prior work and main contributions

As previously mentioned, detection of top-k betweenness centrality nodes in networks is still a challenging problem [43,37,47,29,39–42]. It is obvious that identifying nodes with high betweenness centrality in a large real-world network via computations of centralities for all the nodes is not feasible. However, in most real-world scenarios, the full network topology is not initially known to us. The common approach in this case is to utilize network sampling methods [37,41,42,29,22,44–46]. In these methods, one should collect a subset of nodes as the sample set and then approximate the centralities of all sampled nodes in the induced sub-graph of the network. Finally, the top-k central nodes are selected according to the approximated betweenness values computed for the sampled nodes, and the remaining nodes in the sample set are completely discarded [45]. The latter step is analogous to compression algorithms in which unnecessary information from the data is identified and discarded. The sampling based approaches have three major disadvantages: (1) They induce two sources of error; sampling (collection) error and identification (compression) error. (2) Sampling with complete rate and then removing the least significant estimated values leads to loss of system resources. (3) Proposing algorithms with the capability of sampling with complete rate and direct measurement of network nodes can be difficult, costly and sometimes impossible.

The main focus of this paper is to address the problem of identifying k-highest betweenness centrality nodes in networks while overcoming the aforementioned drawbacks. To this end, we transform this problem into the problem of recovering sparse high-dimensional data from a much smaller number of measurements which is known as sparse signal recovery. The fundamental constraint to make our problem similar to the sparse recovery problem is the *sparsity property* of the desired solution set. Luckily, the problem of this paper satisfies this constraint in which the number of top-k betweenness centrality nodes is much smaller than the set of all network nodes (see Section 3.2). The breakthrough in solving the sparse signal recovery problem is *compressive sensing (CS)* [48–52] which aims to simultaneously sample and compress sparse signals, using *indirect measurements* (see Section 3.3). CS has recently received much attention in several fields (such as astronomy, biology, image and video processing, medicine, and cognitive radio [48]) for its ability to extract sparse information from big data. However, applications of compressive sensing in networks are still in its early stages [53–61], because of some challenging issues (see Section 3.4). One of the most restricting challenges is the construction of *measurement matrix* that needs to be *feasible* based on two fundamental constraints: **(1)** A measurement matrix in networks must contain only non-negative integer elements, which is more restrictive in comparison with random Gaussian measurement matrices usually used in the CS literature. **(2)** Measurements in a network are limited by *network topological constraint*; in other words, only nodes that induce a connected sub-graph can be aggregated together in the same measurement. This is contrary to the assumption of most existing CS results [50] that any subset of vector entries can be aggregated together in the same measurement. We will discuss more in Section 3.3.

In this paper, we introduce **CS-HiBet**, a **C**ompressive **S**ensing based framework for efficiently identifying the k-**Hi**ghest **Bet**weenness centrality nodes without full knowledge of the network topological structure via indirect aggregated measurements. CS-HiBet can perform as a distributed algorithm by letting each node uses only local information on its immediate neighbors.

3. Preliminaries

The main focus of this paper is to address the problem of identifying k-highest betweenness centrality nodes in networks while overcoming the aforementioned drawbacks in Section 2. In this section, after introducing the precise definition of the betweenness centrality (Section 3.1), we show how the problem of detecting top-k highly central nodes satisfies the sparsity property in networks (Section 3.2). Given this, we transform this problem into the problem of recovering sparse high-dimensional data from a much smaller number of measurements which is known as sparse signal recovery. Since the breakthrough in solving the sparse signal recovery problem is *compressive sensing (CS)* [48–50] which addresses the mentioned disadvantages of the sampling methods, it is introduced in Section 3.3. Finally in Section 3.4, we introduce the main challenges in utilizing the strong framework of compressive sensing over networks.

3.1. Network centrality

Consider an undirected network $G = (V, E)$, where V represents the set of nodes (vertices) with cardinality $|V| = n$ and E is the set of links (edges) with cardinality $|E|$. The adjacency matrix of the network G is denoted by Adj. $Adj(u, v) = 1$ if and only if there exists a link between two nodes u and v, and $Adj(u, v) = 0$ otherwise. For a node $u \in V$, we denote its neighbor set as $\mathcal{N}(u) \subset V$, its degree as $deg(u) = |\mathcal{N}(u)|$, and its one-hope adjacency matrix as $egoAdj(u)$.

A key question is how to distinguish the importance of each node in the network. Centrality measures provide the standard means to compare the importance of nodes. Nowadays, centrality measures have become an essential tool for network analysis, and are widely used in diverse applications such as controlling the spread of diseases in a biological network [38], identifying key actors in a terrorist network [62], detecting influential directors in a governance network [8], investigating absence of influential spreaders in rumor dynamics [12], preventing blackouts caused by cascading failure [63], sorting the search results of a search engine [64], cooperative localization in a wireless sensor network [65], and detecting key players and marketing targets in a social network [11,4,13]. The most popular measure as an indicator of a node's importance

is perhaps the *betweenness centrality*. It measures the proportion of shortest paths in the network passing through a specific node. This is the fraction of times that a node acts as a bridge in transferring any valuable information between any pair of nodes along their shortest paths within the network. Hence, this measure is trivially correlated to the nodes' importance from an information-flow standpoint in the network. Let $C_B(u)$ denotes the betweenness centrality of node $u \in V$ which is defined as [6]:

$$C_B(u) = \sum_{v,w,v\neq w} \frac{\sigma_{vw}(u)}{\sigma_{vw}} \tag{1}$$

where σ_{vw} is the total number of shortest paths between every $v, w \in V$, $v \neq w$, and $\sigma_{vw}(u)$ is the number of such paths that pass through node u.

3.2. Sparsity property

Suppose every node $i \in V$ in the network $G = (V, E)$ has a real value x_i (*e.g.* betweenness centrality value over node i), and vector $x = (x_1, x_2, \ldots, x_n)$ is associated with the set V. If $\|x\|_0 = k$, x is a k-sparse vector, namely x has only k non-zero elements in its support, where $\|.\|_0$ is the ℓ_0-norm of a vector. Thus, the sparsity of the vector $x \in \mathcal{R}^n$ is k. When a vector has just a few large coefficients (k) and many small coefficients, it can be well-approximated by a k-sparse vector. It is worth noting that an n-dimensional vector x containing the betweenness centrality of all nodes in a real-world network has the sparsity property, because the number of top-k betweenness centrality nodes is much smaller than the total number of all nodes in the networks ($k \ll n$). For example, Mark Newman [10] showed that the betweenness centrality follows a power law on most networks. The power law as a Long tail distribution suggests that there exist a few nodes with very high betweenness centrality in comparison with the rest of the nodes in the network, which indeed satisfies the sparsity property. As another example, Narayanan [66] investigated several generated genome-wide protein interaction networks for many organisms including Saccharomyces cerevisiae (baker's yeast), Caenorhabditis elegans (worm) and Drosophila melanogaster (fruit fly), and he observed that the distribution of the node betweenness centrality in these networks tends to be a power law. Moreover, Lammer et al. [67] studied the German road network and obtained very broad distributions of betweenness centrality with a power law exponent in the range [1.279, 1.486] (for Dresden = 1.36); These betweenness centrality distributions signal the strong heterogeneity of the network in terms of traffic, with the existence of a very few central road which very probably points to some congestion traffic problems (see Fig. 1). In addition, [68] analyzed a seismic dataset measured in the central zone of Chile before and after the large earthquake of Illapel 2015 considering it as a spatial complex network, and they found a power law betweenness centrality distribution in it. Furthermore, Lee [69] uncovered that the betweenness follows a power law distribution irrespective of the type of networks, and he examined this characteristic in terms of the conditional probability distribution of the betweenness, given the degree. The conditional distribution also exhibits a power law behavior independent of the degree which explains partially, if not whole, the origin of the power law distribution of the betweenness. He then validated this observation on three real networks: the collaboration network, the protein interaction network of D. melanogaster, and the neural network of C. elegans. Besides, the authors in [70] showed that nodes of both fractal and non-fractal scale-free networks have power law betweenness centrality distribution $P(B) \sim B^{-\delta}$, such that for non-fractal scale-free networks $\delta = 2$ and for fractal scale-free networks $\delta = 2 - 1/d$, where d is the dimension of the fractal network. They also supported these results by explicit calculations on four real networks: pharmaceutical firms, yeast, world wide web (WWW), and a sample of the Internet network at autonomous system (AS) level. As a result, the number of top-k betweenness centrality nodes is much smaller than the set of all nodes in a wide range of real-world networks. Therefore, identifying high betweenness centrality nodes in the networks can be framed as recovering sparse high-dimensional data from a much smaller number of measurements, widely known as the *sparse signal recovery* problem.

3.3. Compressive sensing

The breakthrough in solving the sparse signal recovery problem is *compressive sensing* [48–50] which allows for efficiently acquiring and reconstructing a signal, by finding solutions to under-determined linear systems. To be more precise, an n-dimensional sparse signal x can be simultaneously sampled and compressed to a linear sketch y of the original signal, through a measurement matrix $\mathcal{A}_{m\times n}$, where $m \ll n$. Although the dimension of $y_{m\times 1}$ is typically much smaller than that of $x_{n\times 1}$, the sketch y contains plenty of useful information about the data vector x. In this case, the sparsity property is used to uniquely identify and recover the underlying signal x given the measurement vector y and the measurement matrix $\mathcal{A}$. One can formulate this problem by the following linear system with more unknowns than equations which is equivalently under-determined, as:

$$y_{m\times 1} = \mathcal{A}_{m\times n}\, x_{n\times 1} \tag{2}$$

Notice that rank of the matrix $\mathcal{A}_{m\times n}$ is the dimension of the space spanned by its columns, which is equivalent to the dimension of the space spanned by its rows [71]. Therefore, the number of rows or columns of the matrix $\mathcal{A}$ are both trivial upper-bounds for $rank(\mathcal{A})$ and $rank(\mathcal{A}) \leq \min(m, n)$. Consequently, in the case where $m \ll n$ one can conclude:

$$rank(\mathcal{A}) \leq \min(m, n) = m \ll n \tag{3}$$

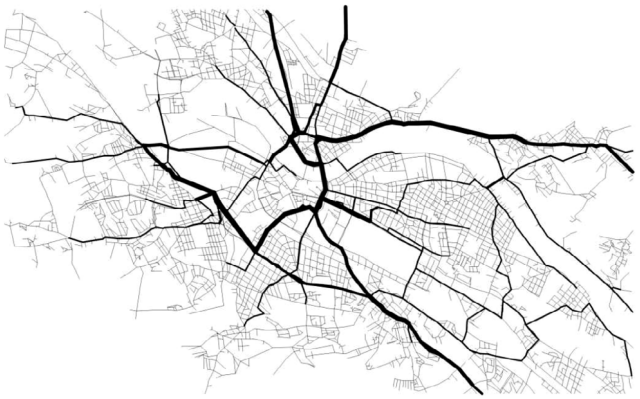


Fig. 1. Betweenness centrality for the road network of Dresden. The width of the links corresponds to the respective betweenness centrality [67].

From the Rank-Nullity theorem [72], we know for an arbitrary matrix $\mathcal{A}$:

$$dimension\big(\mathbb{K}(\mathcal{A})\big) + rank(\mathcal{A}) = n \tag{4}$$

where $\mathbb{K}(\mathcal{A})$ is the kernel or null space of the matrix $\mathcal{A}$ and defined as follows:

$$\mathbb{K}(\mathcal{A}) = \{h \in \mathcal{R}^n : \mathcal{A}h = 0\} \tag{5}$$

From Eqs. (3) and (4), one can deduce that $dimension(\mathbb{K}(\mathcal{A})) > 0$ meaning that there is a non-zero vector present in the kernel of the matrix $\mathcal{A}$, which we refer to as h^*.

Therefore, in case the system $y = \mathcal{A}x$ has a solution x_*, one can introduce infinitely many other solutions in the form $\{x_* + rh^* : r \in \mathcal{R}, r \neq 0\}$, Because:

$$\mathcal{A}(x_* + rh^*) = \mathcal{A}x_* + r\mathcal{A}h^* = \mathcal{A}x_* + r(0) = \mathcal{A}x_* = y \tag{6}$$

This means, without any restricting assumptions, it is not possible to uniquely identify and recover the data vector x via the measurement matrix $\mathcal{A}$ and the measurement vector y [73]. Thus, in order to recover a unique solution to such a system, one must impose extra assumptions. In this regard, compressive sensing adds the sparsity property constraint, allowing only solutions which have a small number of non-zero coefficients as follows [50,49]:

$$\min_x \|x\|_0 \quad \text{s.t} \quad y = \mathcal{A}x \tag{7}$$

It is proved that solving the above optimization problem is NP-hard. Therefore, the sparsity inducing ℓ_1-regularization is considered as a convex relaxation of Eq. (7) as follows:

$$\min_x \|x\|_1 \quad \text{s.t} \quad y = \mathcal{A}x \tag{8}$$

The above objective function is known as *Basis Pursuit* (BP). In addition to the sparsity assumption on x, if the constructed measurement matrix $\mathcal{A}$ satisfies the *Restricted Isometry Property* (RIP), then the recovered answer by BP will be unique.

The k-RIP holds for measurement matrix $\mathcal{A}$ if for all k-sparse vectors x with support $\mathcal{S}$, there exists a *Restricted Isometry Constant* (RIC), $0 < \delta_k < 1$, such that:

$$(1 - \delta_k)\,\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\,\|x\|_2^2 \tag{9}$$

This condition is also equivalent to [74]:

$$\|\widetilde{\mathcal{A}_\mathcal{S}}^\intercal \widetilde{\mathcal{A}_\mathcal{S}} - \mathcal{I}\|_2 \leq \delta \quad s.t. \quad |\mathcal{S}| \leq k \tag{10}$$

where $\mathcal{I}$ is the identity matrix, $\mathcal{A}_\mathcal{S}$ contains all rows of $\mathcal{A}$ but only columns with their indices present in S and ℓ_2-norm of a matrix is equal to its largest singular value. $\widetilde{\mathcal{A}_\mathcal{S}}$ indicates the column-wise normalized $\mathcal{A}_\mathcal{S}$. Eqs. (9) and (10) show that the matrix $\mathcal{A}$ is guaranteed to only change the length of any vector x very little as long as the vector x is k-sparse and has at most k non-zero elements. Note that the vector difference between any two *different* k-sparse vectors is at most $2k$-sparse. The ℓ_2-norm of this difference vector is strictly positive, as otherwise one can conclude that the ℓ_2-norm of this vector is actually zero, which would be a contradiction with the assumption that the two considered k-sparse vectors are different. Given this, if matrix $\mathcal{A}$ satisfies the $2k$-RIP condition, one can deduce that the distance between sketches of two different k-sparse

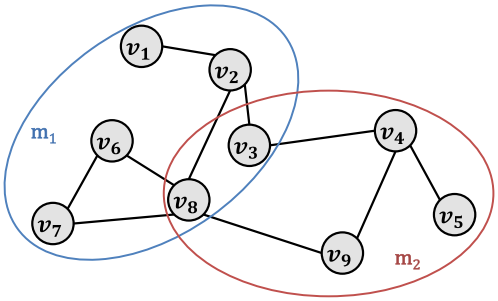


Fig. 2. A network with 9 nodes and 10 links. The measurements m_1 and m_2 are feasible considering the network topological constraints (each of them induces a connected sub-graph over the network).

vectors is non-zero, which will indicate the uniqueness of the recovered answer by the aforementioned convex program with the constraint $y = \mathcal{A}x$. In fact, all methods including Eqs. (7) and (8) will require $\delta_{2k} < 1$ for universal recovery [51,50].

In [75] the authors showed that in case $\delta_{2k} \leq 0.5$ Basis Pursuit is capable of exactly recovering the best k-sparse approximation we were aiming for. Based on this fact, one can formalize the *probability of failure* for recovery over any $\mathcal{S} \subset \{1, 2, \ldots, n\}$ of size $2k$, as follows:

$$\epsilon = P\big[\text{failure for recovery}\big] = P\big[\|\widetilde{\mathcal{A}_{\mathcal{S}}}^{\intercal}\widetilde{\mathcal{A}_{\mathcal{S}}} - \mathcal{I}\|_2 > \delta_{2k} = 0.5\big] \tag{11}$$

The above formulation was used in [76], as well with $k = 1$ to consider specifically. We will present the recovery probabilities for the constructed measurement matrices of our proposed method with the same parameters over different datasets in Section 6.4.5.

Finally note that the strict condition $y = \mathcal{A}x$ within the Basis Pursuit formulation is very sensitive to noise, imperfect sparsity or truncated values in the measurement matrix $\mathcal{A}$, and the sketch y. The following formulation addresses this by removing the exact constraint and penalizing its violation, as:

$$\min_x \|x\|_1 + \|\mathcal{A}x - y\|_2 \tag{12}$$

This objective function is also known as LASSO [77,78] and is used throughout this paper for the optimization step.

3.4. Compressive sensing over networks

Based on the compressive sensing framework, we would like to efficiently recover k highest betweenness centrality nodes from m indirect end-to-end measurements, in a way that $m \ll n$. In the linear system $y_{m\times 1} = \mathcal{A}_{m\times n}\, x_{n\times 1}$, let $\mathcal{A}$ be an $m \times n$ measurement matrix, where its ith row corresponds to the ith feasible measurement. For $i = 1, \ldots, m$ and $j = 1, \ldots, n$, $\mathcal{A}_{ij} = 1$ if and only if node j is visited by the ith measurement, otherwise $\mathcal{A}_{ij} = 0$. Let x be an $n \times 1$ non-negative vector whose jth entry is the value of a certain type of network characteristic (*e.g.* betweenness centrality) over node $j \in V$, and $y \in \mathcal{R}^m$ denotes the measurements vector whose ith entry represents the additive aggregation values of network nodes in the ith row of the measurement matrix $\mathcal{A}$ that induces a *connected sub-graph* over G. Note that this way of measurements construction already satisfies the network topological constraints of the feasibility conditions mentioned at the end of Section 2.

For the example network shown in Fig. 2 with $n = 9$ nodes and $|E| = 10$ links, each of two measurements m_1 and m_2 includes a different subset of connected nodes. The corresponding feasible measurement matrix $\mathcal{A}$ with these measurements is:

$$\mathcal{A} = \begin{array}{c} \\ m_1 \\ m_2 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \tag{13}$$

To understand how the additive aggregation over connected induced sub-graphs is motivated for each measurement in practice, we mention an example from [54]. Consider a network where the nodes represent sensors and the links represent communications between sensors. For the set T of active nodes within an arbitrary feasible measurement that induce a connected sub-graph, a node $u \in T$ monitors the total values corresponding to nodes in T. Every node in T obtains values from its children, if any, and aggregates them with its own value on the spanning tree rooted at u, then sends the sum to its parent. After that, the fusion center can obtain the sum of values corresponding to all the nodes in T by only communicating with u. The explained paradigm in data acquisition and aggregation is highly utilized within the wireless sensor network literature for applications such as air quality monitoring, volcanic activity detection and object localization [79]. Some recent

work has applied a similar acquisition and aggregation paradigm in network tomography [55], community detection [57] and identification of key actors/connections in social networks [58,59].

4. The proposed method: CS-HiBet

In this section, we introduce CS-HiBet, a compressive sensing based approach for efficiently identifying k-highest betweenness centrality nodes in a network. The pseudo code of the proposed method is depicted in Algorithm 1. This algorithm includes 8 steps:

(i) For each node $v \in V$ in the network, its one-hop adjacency matrix with the dimensions of $(\{v\} \cup \mathcal{N}(v)) \times (\{v\} \cup \mathcal{N}(v))$ is constructed. Next, the sum of the reciprocals of the entries in $egoAdj^2(v)\,(1 - egoAdj(v))$ is calculated as the local score for node v. Finally, the probability of selecting a specific node v is also computed as $P_{selection}(v)$, in lines (5)–(9). This pre-processing step can be performed in a parallel or distributed manner by each node independently from the others.
(ii) The first node is selected randomly from the set of all nodes $v_{first} \in V(G)$ in line (13).
(iii) The first node v_{first} is added to the visited set S and all of its neighbors are added to the neighbor set $\mathcal{N}(S)$ in lines (14)–(15).
(iv) The next node is selected relative to $P_{selection}(v)$ for each node $v \in \mathcal{N}(S)$ in line (17).
(v) The selected next node is added to the visited set S and it is removed from the neighbor set $\mathcal{N}(S)$, then its neighbors are added to the neighbor set $\mathcal{N}(S)$, in lines (18)–(20).
(vi) The steps (iv) and (v) are fulfilled 'l' times which is the length of a measurement, to generate a new row for the matrix $\mathcal{A}$ and the vector y in lines (16)–(21).
(vii) The steps (ii)–(v) are repeated 'm' times to construct a feasible measurement matrix $\mathcal{A}$ with 'm' measurements (in parallel) and the corresponding measurement vector y, in lines (10)–(24).
(viii) In order to find the sparse approximation $\hat{x}$ of x, we optimize the LASSO objective function subject to the linear sketch of $y = \mathcal{A}x$, in line (25), based on Eq. (12).

Algorithm 1 The Proposed Method: CS-HiBet

Input: V, m, l
1: V: set of network nodes
2: m: number of measurements
3: l: measurement length
4: $\mathcal{A} =$ NULL ▷ Initializing measurement matrix
5: **Foreach** $v \in V$ **do** ▷ In a parallel or distributed manner
6: $egoAdj(v) =$ The one-hop adjacency matrix of ego node v with the dimensions of $(\{v\} \cup \mathcal{N}(v))^2$
7: $Score(v) =$ The sum of the reciprocals of the entries in $egoAdj^2(v)\,(1 - egoAdj(v))$
8: $P_{selection}(v) = \frac{Score(v)}{\sum_v Score(v)}$
9: **end for**
10: **for** $i = 1 \rightarrow m$ **do** ▷ In a parallel manner
11: $S =$ NULL ▷ Visited set
12: $\mathcal{N}(S) =$ NULL ▷ Neighbor set of visited nodes
13: $v_{first} =$ Select start node randomly from the node set V
14: $S = \{v_{first}\}$ ▷ Add v_{first} to the visited set S
15: $\mathcal{N}(S) = \{u : u \in \mathcal{N}(v_{first})\}$ ▷ Add all neighbors of v_{first} to the $\mathcal{N}(S)$
16: **for** $j = 1 \rightarrow l$ **do**
17: v_{next} = Select next node relative to $P_{selection}(v)$ for $v \in \mathcal{N}(S)$
18: Add v_{next} to the visited set S
19: Remove v_{next} from the neighbor set $\mathcal{N}(S)$
20: Add all neighbors of v_{next} to the neighbor set $\mathcal{N}(S)$
21: **end for**
22: $\mathcal{A}[i, :] =$ Add visited nodes in S to the measurement matrix $\mathcal{A}$ as row
23: $y[i, :] =$ Add the accumulative sum of node values in S to the vector y
24: **end for**
25: $\hat{x} = \min_x \|x\|_1 + \|\mathcal{A}x - y\|_2^2$ ▷ See Equation (12)
Output: sparse approximation $\hat{x}$

Since we want to recover the top-k betweenness centrality nodes, we try to visit these nodes more than the other nodes by our measurements. To achieve this goal, we select the best next node relative to $P_{selection}(v)$. This is for measuring node

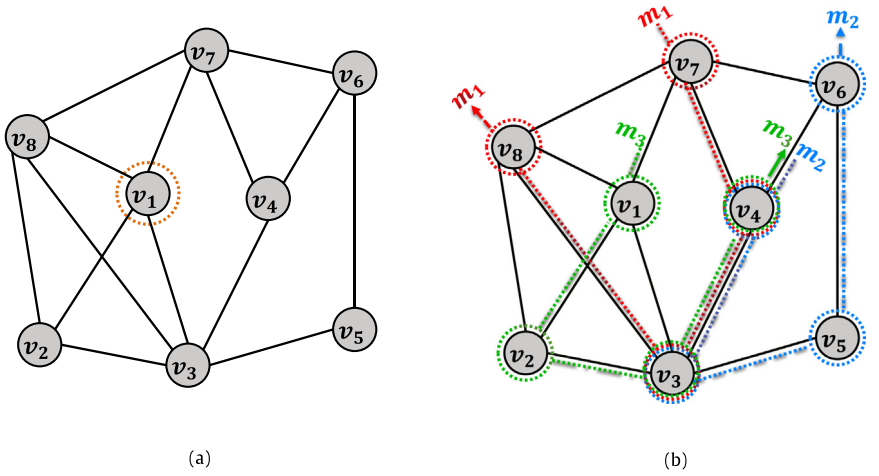


Fig. 3. A network G with 8 nodes and 14 links. (a) The ego node v_1 has 4 nodes in its neighbor set $\mathcal{N}(v_1) = \{v_2, v_3, v_7, v_8\}$. (b) Three measurements m_1, m_2, and m_3 over graph G constructed from the CS-HiBet method.

importance from an information flow standpoint. For the computation of this score, consider node v in the neighbor set $\mathcal{N}(S)$ as ego node, thus every pair of non-adjacent alters must have a geodesic distance of length 2 which passes through ego node v. We only need to consider these geodesics and geodesics of length 1 do not contribute to betweenness centrality. Let $egoAdj(v)$ be the one-hope adjacency matrix of ego node v with the dimensions of $(\{v\}\cup\mathcal{N}(v))\times(\{v\}\cup\mathcal{N}(v))$, then $egoAdj^2_{i,j}(v)$ contains the number of walks of length 2 connecting i and j when $i \neq j$. Therefore, we only need to count the number of walks of length 2 for non-adjacent alters since these will be the geodesics contributing to the local betweenness [80]. It follows that $\left[egoAdj^2(v)\,(1 - egoAdj(v))\right]_{i,j}$ gives the number of geodesics of length 2 joining i to j, where 1 is a matrix of all 1's. The sum of reciprocals of the entries gives the local betweenness of the ego node v.

For instance, consider the network shown in Fig. 3(a), the one-hop adjacency matrix for the ego node v_1 is:

$$egoAdj(v_1) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_7 \\ v_8 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_7 & v_8 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array} \tag{14}$$

Since the matrix is symmetric we only need to consider the zero entries above the leading diagonal and calculates $\left[egoAdj^2(v)\,(1 - egoAdj(v))\right]_{i,j}$ for those entries. These entries are shown by the red box and the calculation gives:

$$egoAdj^2(v_1)\,(1 - egoAdj(v_1)) = \begin{pmatrix} * & * & * & * & * \\ * & * & * & 2 & * \\ * & * & * & 2 & * \\ * & * & * & * & * \\ * & * & * & * & * \end{pmatrix} \tag{15}$$

The local betweenness of the ego node v_1 is then simply the sum of reciprocals of these entries, that is 1, as the $Score(v_1)$. The ego-centric betweenness metric [80] is computationally more tractable than the traditional global betweenness centrality. It can be calculated locally in a parallel or distributed manner, by letting each node communicate only with its immediate neighbors. Then, the probability $P_{selection}(v)$ of selecting a specific node $v \in V$ is calculated by the normalized scores, as in our pre-processing step. The global and local betweenness centrality for all nodes in the sample network in Fig. 3(a) are given in Table 1. This table shows that nodes v_3 and v_4 are central in this network because they have the largest centrality measure. This observation is valid because the number of top-k betweenness centrality nodes is usually much smaller than the total number of all nodes in the networks, as we mentioned in Section 3.2. Hence, one may safely

Table 1
Centrality measures for the sample network in Fig. 3(a). We used the iGraph package in Python to calculate the global betweenness centrality based on Eq. (1). The ego-centric betweenness is calculated based on [80], which is described in this section.

Centrality measure	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
Global betweenness	1.08	0	**6**	**3.5**	1.08	1.08	0.75	1.5
Ego-centric betweenness	1	0	**7**	**4**	1	2	1	2

conclude that these nodes are the most important in the network. We want to recover such central nodes without full knowledge of the network topological structure via indirect aggregated measurements. To this end, CS-HiBet constructs a feasible measurement matrix $\mathcal{A}$ with non-negative integer entries by using m measurements with the step size of l. Every measurement in $\mathcal{A}$ goes through a connected subgraph which guarantees the feasibility of $\mathcal{A}$ considering the network topological constraints. As an example, in the sample network G in Fig. 3(a), we set the number of measurements to 3 ($m = 3$) and the measurement length to 4 ($l = 4$) in the Algorithm 1, then the constructed measurements m_1, m_2, and m_3 are depicted in Fig. 3(b) and the corresponding measurement matrix $\mathcal{A}$ is as follows:

$$\mathcal{A} = \begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \end{array} \begin{array}{c} \begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{16}$$

For the visited nodes in each measurement, the accumulative sum of their scores (ego-centric betweenness) is added to the corresponding element of the measurement vector y. According to the above measurement matrix $\mathcal{A}$ for the sample network Fig. 3(b), the constructed measurement vector is: $y = [14\ 14\ 12]$. To be more precise, the first measurement m_1 is started from v_7 and the walker passes through nodes v_4 and v_3, and it ends with v_8, so the visited node set for m_1 is $S = \{v_3,\ v_4,\ v_7,\ v_8\}$ and its corresponding entry in y is the sum of their scores (ego-centric values in Table 1) which is $y_1 = 7+4+1+2 = 14$. This is similar for m_2 and m_3. After generation of $\mathcal{A}$ and y, we form the linear system of $y_{m\times 1} = \mathcal{A}_{m\times n}\, x_{n\times 1}$. Finally, we find the sparse approximation $\hat{x}$ for this system by optimizing the LASSO objective function, based on Eq. (12). For the constructed $\mathcal{A}$ and y in the network Fig. 3(b), the sparse approximation $\hat{x} = [0.38\ 0.38\ \mathbf{0.61}\ \mathbf{6.61}\ 0.13\ 0.13\ 0.13\ 0.13]$ is the output of CS-HiBet. Thus, one can easily conclude that CS-HiBet detected v_3 and v_4 as the top-2 central nodes in the sample network Fig. 3(b), and it is completely true based on the actual global betweenness centralities in Table 1. We will explore the performance of CS-HiBet via the extensive experimental evaluations in Section 6.

5. Complexity analysis

Consider the network $G = (V, E)$ with the assumption that any arbitrary node $v \in V$ has only local view to its immediate neighbors. The CS-HiBet method can be executed and analyzed in its three main steps according to Algorithm 1.

First, the one-hop adjacency matrix of ego node v, namely $egoAdj(v)$, and also $1 - egoAdj(v)$ can be computed in $O\big(deg(v)^2\big)$, where $deg(v)$ is the degree of node v. Using the famous Strassen's Algorithm, $egoAdj^2(v)$ can be obtained in $O\big(deg(v)^{\log_2^7+o(1)}\big) \approx O\big(deg(v)^{2.8074}\big)$. Finally, at the end of the first step, $Score(v)$ can be locally computed for any arbitrary node $v \in V$ in $O\big(deg(v)^{\log_2^7+o(1)}\big)$ time. Hence, the scores can be locally computed at each node in at most $O\big(\Delta^{\log_2^7+o(1)}\big)$ time, where Δ is the largest degree of a node in the network G. Afterwards, all transition probabilities at each node can also be locally obtained in at most $O(\Delta)$ time.

Second, we can generate m random numbers between 1 and n in $O(m)$ time, to choose the seeds for starting a measurement. m is the number of measurements that correspond to the rows of the measurement matrix $\mathcal{A}$.

Third, we can begin to construct m measurements locally in a distributed manner. For each measurement pre-calculated transition probabilities have been already obtained in the previous steps. They are locally and independently accessible in $O(1)$ time. In the proposed algorithm CS-HiBet, the next node can be determined by using the weighted selection algorithm relative to the probabilities $P_{selection}(u)$ for every $u \in \mathcal{N}(S)$. The selected node u is removed from the $\mathcal{N}(S)$ and added to the set S, then its neighbors are added to the $\mathcal{N}(S)$. The accumulative search for construction of a measurement costs at most $O\big(n\log(n)\big)$ time, by utilizing the binary search. It is noteworthy that at step i from the l total steps corresponding to a measurement, the neighbor set $\mathcal{N}(S)$ contains at most $(i\Delta - i)$ nodes. In this case, each addition and deletion is taking only $O(1)$ time per operation in an array and overall $\min(l\Delta, n)$. The time complexity for the weighted selection search will be

at most $O\big(n\log(n)\big)$ which should be considered in two different cases. One for the case that $l \leq \lfloor \frac{n}{\Delta-1} \rfloor$ and the other is for $l > \lfloor \frac{n}{\Delta-1} \rfloor$. The time complexity in each case is computed as follows:

▷ Case when $l \geq \lfloor \frac{n}{\Delta-1} \rfloor$ then:

$$\begin{aligned}
\text{Search_Cost} &= \sum_{i=1}^{\lfloor \frac{n}{\Delta-1} \rfloor} \log\big(i\Delta - i\big) + \left(n - \left\lfloor \frac{n}{\Delta-1} \right\rfloor\right)\log(n) \\
&= \log\left(\left\lfloor \frac{n}{\Delta-1} \right\rfloor !\right) + \left\lfloor \frac{n}{\Delta-1} \right\rfloor \log\big(\Delta-1\big) + \left(n - \left\lfloor \frac{n}{\Delta-1} \right\rfloor\right)\log(n) \\
&\leq \left\lfloor \frac{n}{\Delta-1} \right\rfloor \log\left(\left\lfloor \frac{n}{\Delta-1} \right\rfloor\right) + \left\lfloor \frac{n}{\Delta-1} \right\rfloor \log\big(\Delta-1\big) + \left(n - \left\lfloor \frac{n}{\Delta-1} \right\rfloor\right)\log(n) \\
&\leq \left\lfloor \frac{n}{\Delta-1} \right\rfloor \left(\log\left(\frac{n}{\Delta-1}\right) + \log\big(\Delta-1\big)\right) + \left(n - \left\lfloor \frac{n}{\Delta-1} \right\rfloor\right)\log(n) \\
&= n\log(n)
\end{aligned}$$

▷ Case when $l < \lfloor \frac{n}{\Delta-1} \rfloor$ then:

$$\begin{aligned}
\text{Search_Cost} &= \sum_{i=1}^{l} \log\big(i\Delta - i\big) \\
&= \log\big(l!\big) + l\log\big(\Delta-1\big) \\
&\leq l\log(l) + l\log(\Delta) \\
&\leq 2n\log(n)
\end{aligned}$$

Therefore, the aggregated total time complexity of the CS-HiBet method will be $O\big(\Delta^{\log_2^7+o(1)} + n\log(n) + m + \min(l\Delta, n)\big)$. As previously mentioned that $m \ll n$ and $\min(l\Delta, n) \leq n$, hence by exploiting the ability of the proposed algorithm to perform locally with only local view at each node, the total time complexity will be $O\big(\Delta^{\log_2^7+o(1)} + n\log(n)\big)$.

The required space storage for any arbitrary node $v \in V$ is $O\big(deg(v)^2\big)$ for the ego adjacency matrix, scores, and the transition probabilities. Thus, the local storage at each node for space complexity of our method will be at most $O(\Delta^2)$. Moreover, the space complexity $O(m)$ is needed to store the randomly chosen initiative seeds and the final measurement vector. Also at most the space $\min(l\Delta, n)$ is needed for the weighted selection algorithm in each measurement.

It is worth noting that in most real-world networks, in particular social networks, nodes are connected to a very small portion of the whole network's nodes, which means Δ is very small. For example, the maximum number of connections allowed on Twitter [81] and Facebook [82] is about 5000 which is much smaller than their network size. Consequently, CS-HiBet can be practically scalable for efficient detection of k-highest betweenness centrality nodes in large real-world networks, in terms of time and space complexity.

6. Experimental evaluation

In this section, we experimentally evaluate the performance of the CS-HiBet, in both real and synthetic networks, under various configurations. First, we introduce the datasets we used for the evaluation. Next, we mention the competing methods we compared our performance to. Then, we explain settings of the tests. Finally, the achieved results and their analysis are shown.

6.1. Datasets

We considered both synthetic and real-world networks for the evaluations. The data from well-known real-world networks were: **(1)** Facebook-like social network [83] with 1899 nodes and 20 296 links; **(2)** Condense Matter Physics (CondMat) network [84] with 23 133 nodes and 93 497 links; **(3)** Twitter's mentions and retweets of the twitter network [85] with 3656 nodes and 188 712 links; **(4)** Arxiv High Energy Physics Theory (HepTh) network [84] with 9877 nodes and 25 998 links; **(5)** Alqaeda terrorist network [86] with 1163 nodes and 18 293 links; **(6)** Wikipedia vote (WikiVote) network [87] with 7115 nodes and 103 689 links; **(7)** Youtube video-sharing network [88] with 11 34 890 nodes and 29 87 624 links; **(8)** Road network of Pennsylvania (RoadNet) [89] with 10 88 092 nodes and 15 41 898 links; **(9)** Pokec social network [90] with 16 32 803 nodes and 306 22 564 links. In case of disconnected networks, we always extracted the largest (strongly) connected component first.

In addition, we used three kinds of synthetic networks: **(10)** Scale-free network based on the Barabási–Albert (BA) model [91] (aka power-law graph) with 500 nodes, 2979 links, average degree of 11.916, and modularity of 0.243, where

links created by each new node were 6; **(11)** Random network based on the Erdös–Rényi (ER) model [92] with 500 nodes, 4000 links, average degree of 16, and modularity of 0.212; **(12)** Small-world network based on the Watts–Strogatz (SW) model [93] with 500 nodes, 4466 links, average degree of 17.864, and modularity of 0.559, where the rewiring probability was 0.2 and the number of initial closest neighbor was 9.

6.2. Competing methods

We refer to our algorithm as CS-HiBet. The baseline methods that we compared our performance to were:

- *DANCE:* In this framework [43], an estimate for a certain centrality metric is approximated based on the *h-neighborhood* of each node using a local function. The h-neighborhood for an arbitrary node v consists of nodes within distance at most h from v in the network. The choice of the local function could be made such that the estimation is tailored towards the desired centrality metric, which makes this framework applicable for ego-centric, efficient, and distributed estimation of several centrality notions.
- *WeightVol:* This method [37] is a model for selecting a set of k nodes as the initial influenced nodes so that they can effectively disseminate the information to the rest of the network from the information flow standpoint. Their proposed method to maximize information diffusion is based on the h-hop node information.
- *LBC:* This method [47] proposed a new centrality metric, called localized bridging centrality, which combines the egocentric betweenness centrality with the locally computable bridging coefficient. This metric can be computed in a distributed way and it performs well in identifying bridging nodes for information flow in the networks.
- *FastApprox:* The authors of [29] proposed an efficient randomized algorithm for betweenness centrality estimation, employing the random sampling of shortest paths, which offers probabilistic guarantees on the quality of the approximation. In order to derive the appropriate sample size necessary to achieve the desired approximation, Vapnik–Chernovenkis (VC) dimension theory [94] is used.
- *K-Path:* [40] introduced a new centrality measure, called k-path, and a randomized algorithm for its estimation. They showed that the nodes with high k-path centrality value have high node betweenness. Their randomized algorithms can estimate the k-path centrality of each node up to additive error of $\pm n^{1/2+\alpha}$ with probability $1 - 1/n^2$. k-path betweenness is based on the random traversal of a message from a certain source similarly to the random-walk betweenness.
- *CS-TopCent:* To address the disadvantages of the sampling-based approaches (see Section 2), [58] proposed a compressive sensing (CS)-based approach for detection of central nodes in networks without full knowledge of the network topology via indirect end-to-end measurements. This method constructs a feasible measurement matrix to efficiently recover top-k central nodes in networks. The measurement matrix construction and the local metric used in CS-TopCent are totally different from what has been proposed in this paper.
- *RW:* Motivated by network tomography problem (indirect measurements), this method, [53], introduced a compressive sensing (CS)-based framework to recover a sparse unknown vector that represents certain features of the elements over the network via collective additive measurements.

Nowadays, researchers commonly deal with large-scale complex networks, so the use of localized information (*i.e.* restricted to a limited h-neighborhood around each node of the network) for centrality-based analysis is gaining momentum in the recent literature. Since all of the aforementioned methods are in this category, it is clear that the obtained results are directly impacted by the choice of h. Although the choice of h is application-specific, the literature (*e.g.* [43,47]) showed that small values, typically $h = 1$ or $h = 2$, yield good results in distributively assessing network centralities for different kinds of complex networks. So, to have a fair comparison with our method in the experiments, we set $h = 1$ for the methods.

6.3. Settings

To evaluate the accuracy of the proposed approach, we measured the *precision* and *recall* of the methods. Precision measures the number of correctly identified nodes (according to the global betweenness centrality values) in the list of k-highest betweenness centrality nodes divided by the total number of identified nodes. Recall measures the number of correctly detected nodes divided by the total number of network nodes. To consider both measures, we used the F-measure metric which represents the harmonic mean of both precision and recall, as:

$$\text{F-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

The CS-HiBet, CS-TopCent, and RW methods possess a source of randomness, hence, in each of their respective experiments we did 10 repetitions. The denoted points in the figures represent the mean value of these repetitions. However, the other implemented competing methods are deterministic and there is no need for repetitions.

The LASSO objective function in Eq. (12), which is used for the optimization step of the CS-HiBet, can be solved extremely quickly using parallel solvers such as POGS [95] by leveraging the power of GPUs. As an example in [96], on a graph with 100 000 nodes and with 10 000 measurements the LASSO objective can be solved in only 21 s on a single Nvidia K40 GPU.

6.4. Evaluation results

We evaluated the performance of the CS-HiBet to detect high betweenness centrality nodes in the networks, based on five different scenarios: (1) Accuracy of CS-HiBet on identifying top-k central nodes for varying percentage of sparsity k, (2) Accuracy of our method on rank prediction for different sparsity levels, (3) Effect of number of measurements m on accuracy, (4) Effect of measurement length l on accuracy, and (5) Recovery probability of the constructed measurement matrix by CS-HiBet for satisfying RIP condition.

6.4.1. Accuracy of CS-HiBet on identifying top-k central nodes

Fig. 4 shows the accuracy evaluation of our approach in comparison with the competing methods, in terms of F-measure for varying sparsity percentage to identify k-highest betweenness centrality nodes. Each point in the horizontal axis is proportional to the number of top-k nodes divided by the number of all nodes in the network (*i.e.* $\frac{k}{n}$). In this experiment, for the CS-based methods (*i.e.* CS-HiBet, CS-TopCent, RW), we performed a set of $m = 0.2n$ measurements of length $l = 0.4n$, in each network. We almost observed an increasing trend for F-measure as we increase the bracket size of top-k nodes. As clearly depicted in all test cases, CS-HiBet performs better than the competing methods in terms of having higher F-measure, even on the lower sparsity (higher value of $\frac{k}{n}$). For higher values of F-measure, one can observe more correlation between the nodes lists identified by the methods and by the global betweenness centrality. Thus, the results show that the CS-HiBet and the betweenness centrality correlate well with regard to the number of correctly identified top-k central nodes.

One of the main reasons for the superiority of the CS-based methods in the accurate detection of top-k betweenness centrality nodes over the h-neighborhood-based competing methods is the fact that: the nodes with a similar h-neighborhood structure will be assigned the same score in each of the latter approaches, however the former approaches may visit those nodes with different rates based on their global positions in the network. As an example, almost all nodes in a line graph have exactly the same 1-hop neighborhood structure resulting in the same assigned scores by each of the latter methods. But, the nodes in the middle of the line graph, which have a higher global betweenness centrality, will have a higher rate of being visited by the measurements (walks) performed in the CS-based approaches. Therefore, these nodes have a higher chance of being recovered as the top-k central nodes.

6.4.2. Accuracy of CS-HiBet on rank prediction

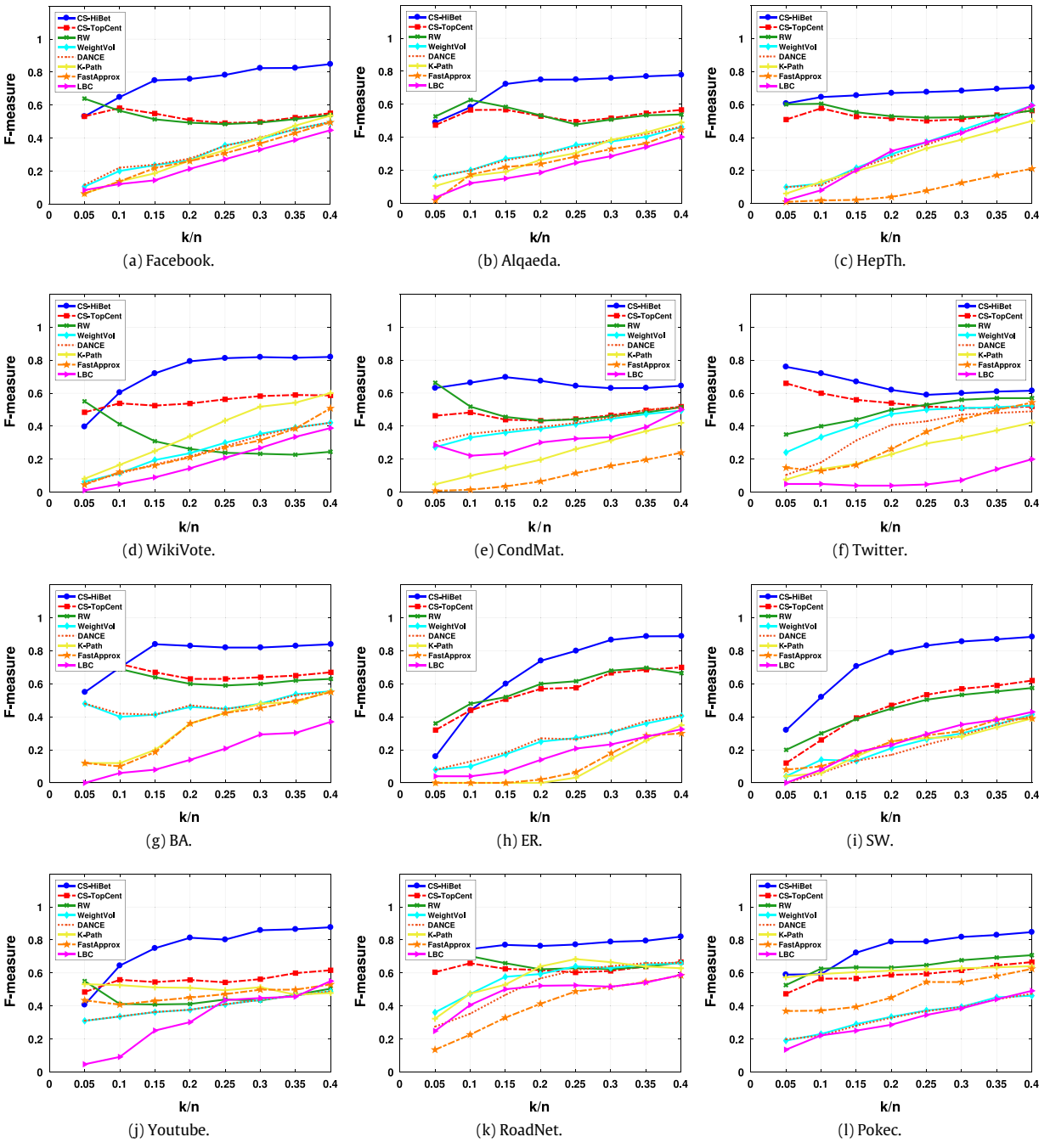
The evaluation described till now focuses on the number of correctly identified nodes in top-k set assembled according to CS-HiBet, DANCE, WeightVol, LBC, K-Path, FastApprox, CS-TopCent, and RW. In a more fine-grained analysis, we are also interested in quantifying the accuracy of ranks assigned by these methods. To this end, we compute the difference between ranks assigned by CS-HiBet and the competing methods, and those determined by the global betweenness centrality. Furthermore, the significance of correct ranking of high-ranked nodes is more important than low-ranked nodes. To address this goal, we consider a distance metric to compare the relevance of two ordered lists. We denote the list of nodes with length k in descending order of importance in the aforementioned methods by $\mathcal{R}_k(\mu)$, and the list of nodes with length k in descending order of the global betweenness centrality by $\mathcal{R}_k(\beta)$. The distance is normalized in the range [0, 1], where 0 corresponds to the perfect match between two given ordered lists, and vice-versa. The normalized weighted distance metric $\mathbf{d} \in [0, 1]$ between these two ordered lists is defined as [97]:

$$\mathbf{d}(\mathcal{R}_k(\mu), \mathcal{R}_k(\beta)) = \frac{\sum_{i \in \mathcal{R}_k(\beta)} \left[\frac{w_i \left| \mathcal{R}_k(\mu_i) - \mathcal{R}_k(\beta_i) \right|}{n - 2i + 1} \right]}{\sum_{i \in \mathcal{R}_k(\beta)} w_i} \tag{18}$$

where w_i is the betweenness of user i and n is the number of nodes in the network. Fig. 5 depicts the accuracy evaluation of the proposed framework CS-HiBet in comparison with the other methods, in terms of the distance between two ordered lists of ranks assigned by them and the global betweenness centrality. Each point in the horizontal axis is proportional to the sparsity level $\frac{k}{n}$. In this experiment, for the CS-based methods (*i.e.* CS-HiBet, CS-TopCent, RW), we performed a set of $m = 0.2n$ measurements of length $l = 0.4n$, in each network. It is clear that in all test cases, CS-HiBet performs better than the competing methods in terms of having lower rank distance with the global betweenness centrality for all sparsity percentages. In addition, we can easily observe an increasing trend for $\mathbf{d}(\mathcal{R}_k(\mu), \mathcal{R}_k(\beta))$ when we increase k. For all datasets at $k = 20\%$ of n, the distance $\mathbf{d}$ for CS-HiBet is lower than 0.17 ($d < 0.17$) and also $d < 0.3$ at $k = 40\%$ of n. Hence, the results show that the CS-HiBet correlates well with the regular betweenness centrality, compared to the competing methods, in terms of estimated ranks for top-k central nodes.

6.4.3. Effect of number of measurements m on accuracy

In this test, we show that CS-HiBet is capable of achieving a higher F-measure with a lower required *number of measurements* in comparison with the CS-based competing methods (*i.e.* CS-TopCent and RW). It is worth noting that the concept of number of measurements is irrelevant for the rest of the aforementioned competing methods. In Fig. 6, each point in the horizontal axis corresponds to the number of required measurements m divided by the number of all nodes n in the



(a) Facebook. (b) Alqaeda. (c) HepTh.

(d) WikiVote. (e) CondMat. (f) Twitter.

(g) BA. (h) ER. (i) SW.

(j) Youtube. (k) RoadNet. (l) Pokec.

Fig. 4. Comparison of accuracy between CS-HiBet and the competing methods for the number of correctly identified top-k betweenness centrality nodes in networks for varying sparsity percentage. For higher values of F-measure, one can observe more correlation between the nodes lists identified by the methods and by the global betweenness centrality.

network (*i.e.* $\frac{m}{n}$). For all datasets, we set the length of measurements to $l = 0.2n$ and the sparsity to $k = 0.15n$. In each of the test cases for the datasets, we did 10 repetitions. The denoted points in the figures represent the mean value of these repetitions along with their asymmetric standard deviations, which quantifies the amount of variations of F-measure at each point in each figure. It is clear that in all test cases, CS-HiBet outperforms the two other methods in terms of having higher F-measure for most number of measurements. The average improvements of our method in comparison with CS-TopCent and RW on all datasets are around 31% and 48%, respectively.

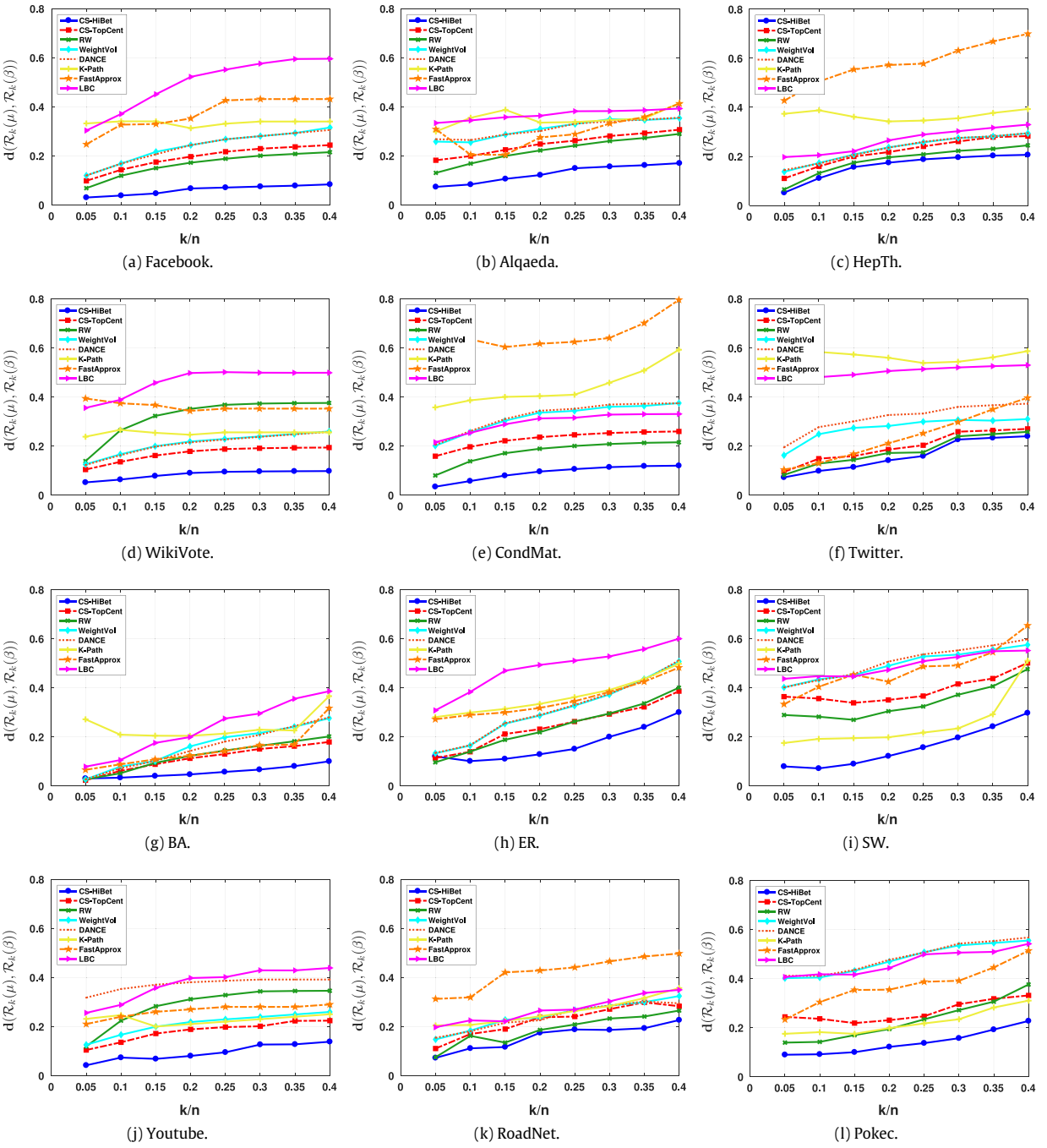
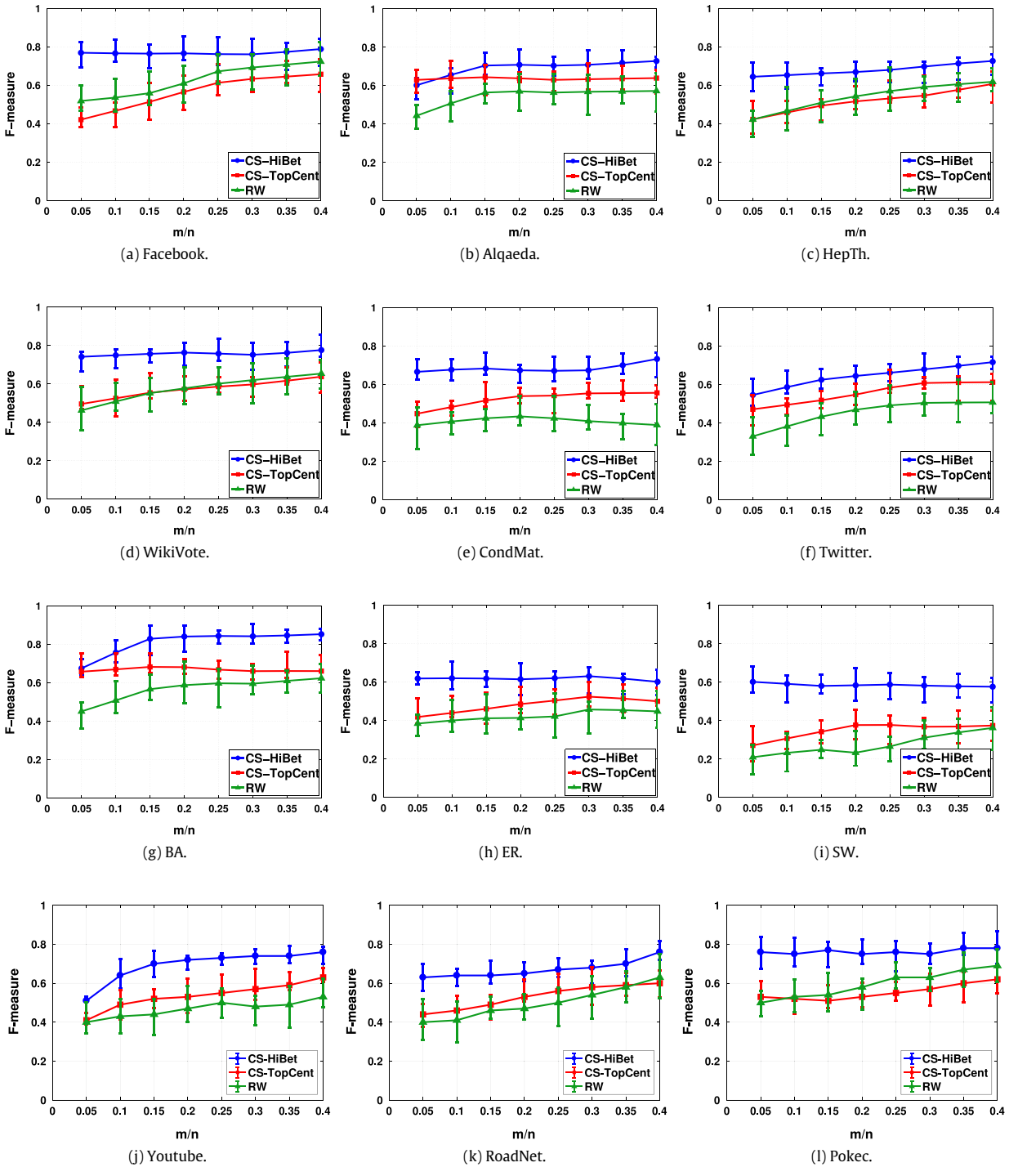


Fig. 5. Comparison of accuracy for the distance between top k ranks assigned by CS-HiBet and the competing methods ($\mathcal{R}_k(\mu)$), and those determined by the global betweenness centrality ($\mathcal{R}_k(\beta)$). The lower the value of $\mathbf{d}(\mathcal{R}_k(\mu), \mathcal{R}_k(\beta))$ is, the better match between the ordered lists identified by the methods and the betweenness centrality will be observed.

6.4.4. Effect of measurement length l on accuracy

This test demonstrates that CS-HiBet achieves a higher F-measure with a lower *measurement length* compared to the CS-based competing methods (*i.e.* CS-TopCent and RW). Note that the concept of measurement length is irrelevant for the rest of the aforementioned competing methods. In Fig. 7, each point in the horizontal axis corresponds to the measurement length l divided by the number of all nodes n in the network (*i.e.* $\frac{l}{n}$). This experiment is conducted under the situation that the sparsity sets to $k = 0.15n$ and the total number of measurements sets to $m = 0.2n$ for each dataset. The denoted points



(a) Facebook. (b) Alqaeda. (c) HepTh.

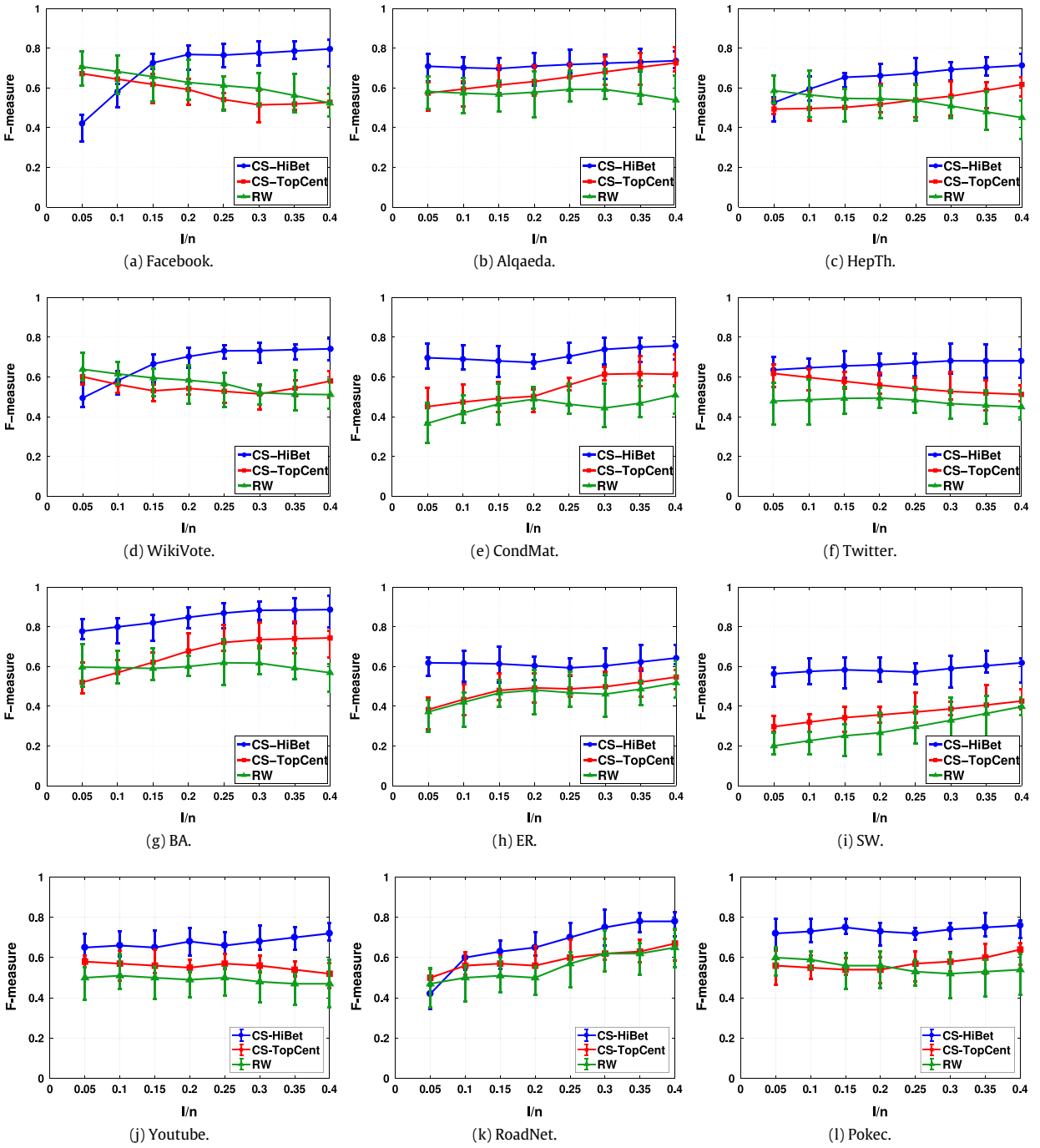
(d) WikiVote. (e) CondMat. (f) Twitter.

(g) BA. (h) ER. (i) SW.

(j) Youtube. (k) RoadNet. (l) Pokec.

Fig. 6. Effect of number of measurements m on the accuracy of CS-HiBet in comparison with CS-TopCent and RW in terms of F-measure. For all networks, we performed the measurements of length $0.2n$ and the sparsity sets to $0.15n$.

in the figures represent the mean value of 10 repetitions with their asymmetric standard deviations. As illustrated in Fig. 7, the CS-HiBet has higher F-measure for the most measurement lengths in all test cases. The average improvements of our method in comparison with CS-TopCent and RW on all datasets are around 28% and 41%, respectively.



(a) Facebook. (b) Alqaeda. (c) HepTh.

(d) WikiVote. (e) CondMat. (f) Twitter.

(g) BA. (h) ER. (i) SW.

(j) Youtube. (k) RoadNet. (l) Pokec.

Fig. 7. Effect of measurement length l on the accuracy of CS-HiBet compared to CS-TopCent and RW in terms of F-measure. For all networks, we performed $0.2n$ measurements and the sparsity sets to $0.15n$.

6.4.5. Recovery probability

Using the definitions and parameters presented in Section 3.3, the probability of recovery based on Eq. (11) can be defined as:

$$P[\text{recovery}] = \frac{\left|\{S : \|\mathcal{A}_S{}^{\intercal}\mathcal{A}_S - \mathcal{I}\|_2 \leq 0.5, |S| \leq 2\}\right|}{\left|\{S : |S| \leq 2\}\right|} \tag{19}$$

Table 2
Recovery probability for the constructed measurement matrix in CS-Hi-Bet for different data-sets.

Network	Facebook	Alqaeda	HepTh	WikiVote	CondMat	Twitter
Recovery Prob.	0.771	0.858	0.754	0.796	0.861	0.757
Network	BA	ER	SW	Youtube	RoadNet	Pokec
Recovery Prob.	0.963	0.989	0.997	0.775	0.801	0.742

In Table 2, the recovery probabilities for the constructed measurement matrices are reported on all previously introduced synthetic and real datasets, using the same measurement matrices utilized in Section 6.4.1. The results show that the constructed measurement matrix via the CS-HiBet poses a high recovery probability in the aforementioned networks.

7. Conclusion

Betweenness centrality has been widely used as a fundamental metric for quantitatively measuring the relative importance of nodes in a network. It is highly correlated to the impact of a specific node on the spread of influence in social networks, the user activity in mobile phone networks, the contagion process in biological networks, and bottlenecks in communication networks. Thus, identification of k-highest betweenness centrality nodes is of great interest. Although many exact and approximation schemes have been proposed for this problem, the vast majority of these algorithms fail to scale on real-world networks because of their high time and space complexity. On the other hand, some of them tend to assume full knowledge of the network topological structure which is not often the case in real networks. Another fact that needs to be taken into consideration is that direct measurement of each individual node in networks may impose remarkable overhead. To overcome these shortcomings, we proposed CS-HiBet, a novel approach for efficiently identifying top-k betweenness centrality nodes in networks, using compressive sensing with indirect end-to-end measurements. We assumed that each node has only localized information about its neighbors allowing our approach to perform as a distributed algorithm. Extensive experimental evaluations on synthetic and real datasets demonstrated that the CS-HiBet and global betweenness centrality correlate very well with regard to the number of correctly identified central nodes and their estimated rank in networks. Moreover, the results indicated the efficiency of CS-HiBet on the required number of measurements (m) and their lengths (l). In addition, CS-HiBet as a localized algorithm is time and space efficient for utilization in real-world networks.

References

[1] S.H. Strogatz, Exploring complex networks, Nature 410 (2001) 268–276.
[2] S. Dorogovtsev, J.F.F. Mendes, Evolution of networks, Adv. Phys. 51 (2002) 1079–1187.
[3] S. Xu, P. Wang, Identifying important nodes by adaptive LeaderRank, Physica A 469 (2017) 654–664.
[4] L. Lu, T. Zhou, Q.-M. Zhang, H. Stanley, The h-index of a network node and its relation to degree and coreness, Nature Commun. 7 (2016) 10168.
[5] J. Bae, S. Kim, Identifying and ranking influential spreaders in complex networks by neighborhood coreness, Physica A 395 (2014) 549–559.
[6] L. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1977) 35–41.
[7] D.-B. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, T. Zhou, Identifying influential nodes in complex networks, Physica A 391 (2012) 1777–1787.
[8] X. Huang, I. Vodenska, F. Wang, S. Havlin, H.E. Stanley, Identifying influential directors in the United States corporate governance network, Phys. Rev. E 84 (2011) 046101.
[9] L.D.F. Costa, F.A. Rodrigues, G. Travieso, P.R. Villas-Boas, Characterization of complex networks: A survey of measurements, Adv. Phys. 56 (2007) 167–242.
[10] S. Newman, Networks: An Introduction, Oxford University Press, 2010, pp. 168–234.
[11] M. Lee, S. Choi, C. Chung, Efficient algorithms for updating betweenness centrality in fully dynamic graphs, Inform. Sci. 326 (2016) 278–296.
[12] J. Borge-Holthoefer, Y. Moreno, Absence of influential spreaders in rumor dynamics, Phys. Rev. E 85 (2012) 026116.
[13] S.P. Borgatti, Identifying sets of key players in a social network, Comput. Math. Organ. Theory 12 (2006) 21–34.
[14] M. Everett, T. Valente, Bridging, brokerage and betweenness, Social Networks 44 (2016) 202–208.
[15] G. Kahng, E. Oh, B. Kahng, D. Kim, Betweenness centrality correlation in social networks, Phys. Rev. E 67 (2003) 017101.
[16] M. Ortiz, J. Hoyos, M. Lopez, The social networks of academic performance in a student context of poverty in Mexico, Social Networks 26 (2) (2004) 175–188.
[17] Y. Said, E. Wegman, W. Sharabati, J. Rigsby, Social networks of author-coauthor relationships, Comput. Statist. Data Anal. 52 (2008) 2177–2184.
[18] H. Jeong, S. Mason, A. Barabasi, Z. Oltvai, Lethality and centrality in protein networks, Nature 411 (2001) 41–42.
[19] L.A. Maglaras, D. Katsaros, New measures for characterizing the significance of nodes in wireless ad hoc networks via localized path-based neighborhood analysis, Soc. Netw. Anal. Min. 2 (2011) 97–106.
[20] S. Catanese, E. Ferrara, G. Fiumara, Forensic analysis of phone call networks, Soc. Netw. Anal. Min. 3 (2013) 15–33.
[21] C.S. Ang, Interaction networks and patterns of guild community in massively multiplayer online games, Soc. Netw. Anal. Min. 1 (4) (2011) 341–353.
[22] E. Bergamini, H. Meyerhenke, Approximating betweenness centrality in fully dynamic networks, Internet Math. 12 (5) (2016) 281–314.
[23] S. Macskassy, Contextual linking behavior of bloggers: leveraging text mining to enable topic-based analysis, Soc. Netw. Anal. Min. 1 (4) (2011) 355–375.
[24] A. Zhao, B. Zhao, Y. Cui, A network centrality measure framework for analyzing urban traffic flow: A case study of Wuhan, China, Physica A 478 (2017) 143–157.
[25] B. Singh, N. Gupte, Congestion and decongestion in a communication network, Phys. Rev. E 71 (5) (2005) 055103.
[26] S.M. Taheri, H. Mahyar, M. Firouzi, E. Ghalebi K., R. Grosu, A. Movaghar, Extracting implicit social relation for social recommendation techniques in user rating prediction, in: Social Computing Workshop: Spatial Social Behavior Analytics on the Web At 26th International World Wide Web Conference (WWW), 2017.

[27] E. Bergamini, P. Crescenzi, G. D'Angelo, H. Meyerhenke, L. Severini, Y. Velaj, Improving the betweenness centrality of a node by adding links, 2017, pp. 1–28, arXiv:1702.05284v1.
[28] U. Brandes, A faster algorithm for betweenness centrality, J. Math. Sociol. 25 (2001) 163–177.
[29] M. Riondato, E.M. Kornaropoulos, Fast approximation of betweenness centrality through sampling, Data Min. Knowl. Discov. 30 (2016) 438–475.
[30] M. Borassi, E. Natale, KADABRA is an ADaptive algorithm for betweenness via random approximation, in: Annual European Symposium on Algorithms, ESA, 2016, pp. 1–20.
[31] P. Pantazopoulos, M. Karaliopoulos, I. Stavrakakis, On the local approximations of node centrality in internet router-level topologies, Self-Organizing Systems 8221 (2014) 115–126.
[32] M. Riondato, E. Upfal, ABRA: Approximating betweenness centrality in static and dynamic graphs with rademacher averages, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD, 2016, pp. 1145–1154.
[33] P. Wisniewski, B. Knijnenburg, H. Lipford, Making privacy personal: Profiling social network users to inform privacy education and nudging, Int. J. Hum.-Comput. Stud. 98 (2017) 95–108.
[34] Facebook, Facebook statement of rights and responsibilities, 2015, URL http://www.facebook.com/terms.php.
[35] G. Sabidussi, The centrality index of a graph, Psychometrika 31 (1966) 581–603.
[36] Y. Yao, D. Liao, Identifying all-around nodes for spreading dynamics in complex networks, Physica A 391 (2012) 4012–4017.
[37] H. Kim, E. Yoneki, Influential neighbours selection for information diffusion in online social networks, in: International Conference on Computer Communications and Networks, ICCCN, 2012.
[38] J.G. Liu, Z.M. Ren, Q. Guo, Ranking the spreading influence in complex networks, Physica A 392 (2013) 4154–4159.
[39] K. Avrachenkov, N. Litvak, D. Nemirovsky, E. Smirnova, M. Sokol, Monte Carlo methods for top-k personalized pagerank lists and name disambiguation, Tech. Report RR-7367, INRIA, 2010, pp. 1–29.
[40] N. Kourtellis, T. Alahakoon, R. Simha, A. Iamnitchi, R. Tripathi, Identifying high betweenness centrality nodes in large social networks, Soc. Netw. Anal. Min. 3 (2013) 899–914.
[41] S. Ji, Z. Yan, Refining approximating betweenness centrality based on samplings, 2017, pp. 1–13, arXiv:1608.04472v5.
[42] F. Bonchi, G. De Francisci Morales, M. Riondato, Centrality measures on big graphs: Exact, approximated, and distributed algorithms, in: International Conference Companion on World Wide Web (WWW), 2016, pp. 1017–1020.
[43] K. Wehmuth, A.T.A. Gomes, A. Ziviani, DANCE: A framework for the distributed assessment of network centralities, 2014, pp. 1–12, arXiv:1108.1067v2.
[44] P. Wang, J. Zhao, B. Ribeiro, J.C. Lui, D. Towsley, X. Guan, Practical characterization of large networks using neighborhood information, 2013, pp. 1–12, arXiv:1311.3037v1.
[45] A.S. Maiya, T.Y. Berger-Wolf, Online sampling of high centrality individuals in social networks, Adv. Knowl. Discov. Data Min. 6118 (2010) 91–98.
[46] Y. Lim, D.S. Menasche, B. Ribeiro, D. Towsley, P. Basu, Online estimating the k central nodes of a network, in: IEEE Network Science Workshop, 2011, pp. 118–122.
[47] S. Nanda, D. Kotz, Localized bridging centrality for distributed network analysis, in: International Conference on Computer Communications and Networks, ICCCN, 2008, pp. 1–6.
[48] M. Davenport, M. Duarte, Y. Eldar, G. Kutyniok, Introduction to compressed sensing, in: Compressed Sensing: Theory and Applications, Cambridge University Press, 2012, pp. 1–64.
[49] E.J. Candes, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure Appl. Math. 59 (8) (2006) 1207–1223.
[50] D. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (4) (2006) 1289–1306.
[51] E.J. Candes, T. Tao, Decoding by linear programming, IEEE Trans. Inform. Theory 51 (12) (2005) 4203–4215.
[52] E.J. Candes, Near-optimal signal recovery from random projections: Universal encoding strategies, IEEE Trans. Inform. Theory 52 (12) (2006) 5406–5425.
[53] W. Xu, E. Mallada, A. Tang, Compressive sensing over graphs, in: IEEE INFOCOM, 2011, pp. 2087–2095.
[54] M. Wang, W. Xu, E. Mallada, A. Tang, Sparse recovery with graph constraints: fundamental limits and measurement construction, in: IEEE INFOCOM, 2012, pp. 1871–1879.
[55] H. Mahyar, H.R. Rabiee, Z.S. Hashemifar, UCS-NT: An unbiased compressive sensing framework for network tomography, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, Vancouver, Canada, 2013, pp. 4534–4538.
[56] H. Mahyar, H.R. Rabiee, Z.S. Hashemifar, P. Siyari, UCS-WN: An unbiased compressive sensing framework for weighted networks, in: Conference on Information Sciences and Systems, CISS, Baltimore, USA, 2013, pp. 1–6.
[57] H. Mahyar, H.R. Rabiee, A. Movaghar, E. Ghalebi, A. Nazemian, CS-ComDet: A compressive sensing approach for inter-community detection in social networks, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, Paris, France, 2015, pp. 89–96.
[58] H. Mahyar, Detection of top-k central nodes in social networks: A compressive sensing approach, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, Paris, France, 2015, pp. 902–909.
[59] H. Mahyar, H.R. Rabiee, A. Movaghar, R. Hasheminezhad, E. Ghalebi, A. Nazemian, A low-cost sparse recovery framework for weighted networks under compressive sensing, in: IEEE International Conference on Social Computing and Networking, SocialCom, Chengdu, China, 2015. pp. 183–190.
[60] E. Ghalebi K., H. Mahyar, R. Grosu, H.R. Rabiee, Compressive sampling for sparse recovery in networks, in: The 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD, 13th MLG Workshop, Halifax, Nova Scotia, Canada, 2017.
[61] H. Mahyar, E. Ghalebi K., H.R. Rabiee, R. Grosu, The bottlenecks in biological networks, in: the 34th International Conference on Machine Learning, ICML, Computational Biology Workshop, Sydney, Australia, 2017.
[62] I. Hamed, M. Charrad, Recognizing information spreaders in terrorist networks: 26/11 attack case study, in: Lecture Notes in Business Information Processing, vol. 233, 2015, pp. 1–12.
[63] A.E. Motter, Y.C. Lai, Cascade-based attacks on complex networks, Phys. Rev. E 6 (2002) 065102.
[64] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. ISDN Syst. 30 (1998) 107–117.
[65] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, IEEE Signal Process. Mag. 22 (2005) 54–69.
[66] S. Narayanan, The betweenness centrality of biological networks, University Libraries, Virginia Polytechnic Institute and State University, 2005, URL https://books.google.com/books?id=KZ26DAEACAAJ.
[67] S. Lämmer, B. Gehlsen, D. Helbing, Scaling laws in the spatial structure of urban road networks, Physica A 363 (1) (2006) 89–95.
[68] D. Pastén, F. Torres, B. Toledo, V. Muñoz, J. Rogan, J.A. Valdivia, Time-based network analysis before and after the M_w 8.3 Illapel Earthquake 2015 Chile, Pure Appl. Geophys. 173 (7) (2016) 2267–2275.
[69] C.-Y. Lee, Correlations among centrality measures in complex networks, 2006, preprint arxiv:physics/0605220, 2006.
[70] M. Kitsak, S. Havlin, G. Paul, M. Riccaboni, F. Pammolli, H.E. Stanley, Betweenness centrality of fractal and nonfractal scale-free model networks and tests on real networks, Phys. Rev. E 75 (5) (2007) 056115.
[71] G. Mackiw, A note on the equality of the column, and row rank of a matrix, Math. Mag. 68 (4) (1995) 285.
[72] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, 2000.

[73] G.V. Shenoy, Linear Programming: Methods and Applications, 2007.
[74] S. Foucart, H. Rauhut, A Mathematical Introduction to Compressive Sensing, Vol. 1, 2013.
[75] T.T. Cai, A. Zhang, Sharp RIP bound for sparse signal and low-rank matrix recovery, Appl. Comput. Harmon. Anal. 35 (1) (2013) 74–93.
[76] D.C. Dhanapala, V.W. Bandara, A. Pezeshki, A.P. Jayasumana, Phenomena discovery in WSNs: A compressive sensing based approach, in: Communications, ICC, 2013 IEEE International Conference on, 2013, pp. 1851–1856.
[77] R. Tibshirani, Regression shrinkage and selection via the LASSO, J. R. Stat. Soc. Ser. B 58 (1994) 267–288.
[78] E.J. Candes, M. Rudelson, T. Tao, R. Vershynin, Error correction via linear programming, in: 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS, 2005, pp. 668–681.
[79] R. Middya, N. Chakravarty, M.K. Naskar, Compressive sensing in wireless sensor networks-a survey, IETE Technical Review, 2016.
[80] M. Everett, S.P. Borgatti, Ego network betweenness, Social Networks 27 (2005) 31–38.
[81] Twitter Connections Limit, 2017, https://support.twitter.com/articles/66885.
[82] Facebook Connections Limit, 2013, https://www.facebook.com/help/community/question/?id=492434414172691.
[83] T. Opsahl, P. Panzarasa, Clustering in weighted networks, Social Networks 31 (2) (2009) 155–163.
[84] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data (TKDD) 1 (1) (2007) 2.
[85] Twitter, Gephi platform for interactive visualization and exploration of graphs, 2017, URL http://rankinfo.pkqs.net/twittercrawl.dot.gz.
[86] InfoPath, Stanford network analysis platform, 2017, URL http://snap.stanford.edu/infopath.
[87] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: WWW, 2010.
[88] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowl. Inf. Syst. 42 (1) (2015) 181–213.
[89] J. Leskovec, K.J. Lang, A. Dasgupta, M.W. Mahoney, Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, Internet Math. 6 (1) (2009) 29–123.
[90] L. Takac, M. Zabovsky, Data analysis in public social networks, in: International Scientific Conference and International Workshop Present Day Trends of Innovations, vol. 1, 2012.
[91] A.L. Barabasi, R. Albert, Emregence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[92] P. Erdos, A. Renyi, On the evolution of random graphs, Publication of the Mathematical Institute of the Hungarian Academy of Science, 1960, pp. 17–61.
[93] D.J. Watts, S.H. Strogatz, Collective dynamics of small-world networks, Nature 393 (6684) (1998) 440–442.
[94] V.N. Vapnik, A.Y. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, in: Measures of Complexity, Springer, 2015, pp. 11–30.
[95] POGS, Proximal operator graph solver, 2017, URL http://foges.github.io/pogs/.
[96] N. Parikh, S. Boyd, Block splitting for distributed optimization, Math. Program. Comput. 6 (1) (2014) 77–102.
[97] M.U. Ilyas, M.Z. Shafiq, A.X. Liu, H. Radha, A distributed algorithm for identifying information hubs in social networks, IEEE J. Sel. Areas Commun. 31 (9) (2013) 629–640.