Compressive Sampling for Sparse Recovery in Networks

Elahe Ghalebi K.[1]
Department of Computer Engineering,
Vienna University of Technology
elahe.ghalebi@tuwien.ac.at

Hamidreza Mahyar[1*]
Department of Science and Engineering,
Sharif University of Technology
hmahyar@ce.sharif.edu

Radu Grosu
Department of Computer Engineering,
Vienna University of Technology
radu.grosu@tuwien.ac.at

Hamid R. Rabiee
Department of Computer Engineering,
Sharif University of Technology
rabiee@sharif.edu

ABSTRACT

This paper introduces CS-SRN, a general framework for recovering sparse vectors representing specific features of nodes/links in networks. We use compressive sampling (CS) to construct a feasible measurement matrix under network topological constraints which is motivated by network inference. CS-SRN addresses the problem of monitoring the network internal characteristics using indirect end-to-end (aggregated) measurements. We evaluate the performance of the proposed method by extensive simulations on both synthetic and real-world networks under several configurations. The experimental results indicate that this framework outperforms the state-of-the-art compressive sensing-based method and can be employed to efficiently and accurately infer a wide range of networks.

CCS CONCEPTS

• **Human-centered computing** → **Social network analysis**; • **Networks** → *Network monitoring*; • **Theory of computation** → *Graph algorithms analysis*;

KEYWORDS

Compressive Sampling, Sparse Recovery, Network Tomography, Graph Mining

ACM Reference format:
Elahe Ghalebi K.[1], Hamidreza Mahyar[1], Radu Grosu, and Hamid R. Rabiee. 2017. Compressive Sampling for Sparse Recovery in Networks. In *Proceedings of International Workshop on Mining and Learning with Graphs, Halifax, Nova Scotia, Canada, August 2017 (MLG'17),* 8 pages.
https://doi.org/10.475/123_4xxxxx

* A preliminary version of our results appears in [27].
[1] Authors contributed equally

MLG'17, August 2017, Halifax, Nova Scotia, Canada

ACM ISBN 123-4567-24-567/08/06xxxxx. . . $15.00
https://doi.org/10.475/123_4xxxxx

1 INTRODUCTION

A very wide range of real-world systems can be modeled and structured by the means of networks (graphs), where actors of the system are indicated by nodes (vertices), and the existing connections between actors are demonstrated by links (edges). The common examples of such networks include technological and transportation infrastructures, communication systems, biological systems, information systems, and a variety of social interaction structures [17, 32]. Since most of these networks are growing so fast and emerging toward more decentralized management, the importance of network monitoring (*i.e.* QoS measurement, troubleshooting, and service-level verification) has magnified in recent years. Monitoring is one of the most challenging tasks in most of the networks. The approaches that closely rely on direct measurements or cooperation of individual nodes are not always possible or economic due to several limitations, such as protocols and/or topological constraints of networks. Thus, the monitoring of network internal characteristics using indirect end-to-end (aggregated) measurements seems an essential task in network inference, which is known as *Network tomography* [6, 12, 18, 41].

Often, it is very beneficial to exploit the features of each individual node/link with a total number of indirect aggregated measurements much smaller than the number of all nodes/links in the network. This is conceivable if we know about the sparse nature of these features. For example in a computer network, the number of congested links is small relative to all links, in a biological network, the number of infectious diseases hubs is much smaller than the set of all nodes, or in a social network, the number of influential spreaders and marketing targets is relatively small to all nodes.

In this paper, we introduce "CS-SRN", a compressive sampling framework for efficiently recovering the sparse structures in networks. Compressive sampling (CS) [5, 7, 9, 15, 16] is a new research domain in signal processing and information theory which would like to recover high-dimensional sparse signals from a much smaller non-adaptive linear measurements or incomplete observations. Its goal is to sample and compress sparse signals, simultaneously. The preliminary idea of CS [9] is that in a proper lower dimensional space, the under-sampled representation of a signal covers the most significant information about that signal. The fundamental challenge in CS is to correctly recover the data vectors up to a certain sparsity percentage.

Compressive sampling is of great interest to various research communities, such as communication theory [27, 28, 30], biological systems [21], sensor networks [31], and social networks [26, 29]. One of the most important challenges in applying CS on networks (graphs) is the construction of a feasible measurement matrix which should satisfy two essential constraints:

(1) In general CS applications, random Gaussian measurement matrices usually used, but in networking, every measurement should contain only non-negative integers.
(2) Most existing work in CS assumes that any subset of vector entities can be aggregated together in the same measurement [9, 15], but it is not a realistic assumption in the network monitoring problem, because only nodes/links in a connected sub-graph can be considered as one measurement. We call this limitation as *network topological constraints*.

As a result, using compressive sampling in network inference needs entirely different assumptions in comparison with other CS applications, so it is interesting in its own right. There have been a few work for the construction of a feasible measurement matrix in networks (graphs) [13, 20, 24, 26–30, 37, 40]. In general, we can categorize these approaches into either deterministic or random methods. The work in [40] (RW) is one of the state-of-the-art CS-based algorithms for sparse recovery in networks, which has three major drawbacks: (1) relatively high number of required measurements, (2) low accuracy in detecting the internal features vectors of the network, and (3) significant linear bias towards high-degree nodes for node selection in each measurement, which make it inapplicable for many real-world networks. In other words, this method can mostly recover sparse specifications of nodes/links that are pointing to the neighbors of large degree nodes. To tackle these problematic issues, in this paper, we propose a general framework in the context of compressive sampling for network inference by efficiently extracting the link sparse specifications in the networks. We experimentally evaluated the proposed method in several synthetic and real datasets under various configurations. The results demonstrate that our method has a significant improvement compared to RW.

2 MODEL AND PROBLEM FORMULATION

We express a network by graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes (vertices) with cardinality $|V| = n$, and $E = \{e_1, e_2, ..., e_{|E|}\}$ is the set of links (edges) with cardinality $|E|$. Let $deg(v)$ be the degree of a node $v \in V$ and $\mathcal{N}(v) \subset V$ be the list of its neighbors. Suppose every link i has a real value x_i, and vector $x = (x_i, i = 1, 2, ..., |E|)$ is associated with the link set E. x is a k-sparse vector if and only if $\|x\|_0 = k$, where $\|.\|_0$ denotes the number of non-zero elements in the support of x. Suppose we have m end-to-end

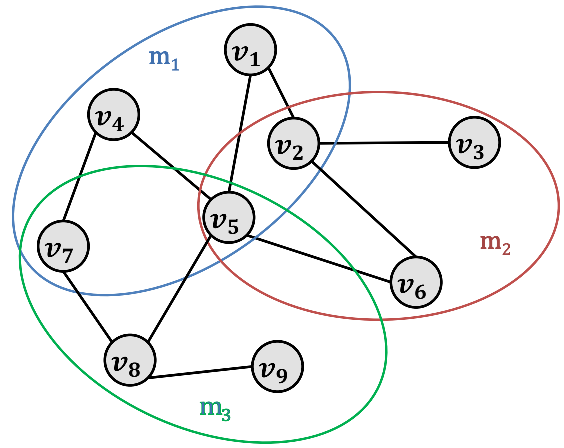


Figure 1: A network with 9 nodes, 10 links, and 3 feasible measurements

measurements over the network ($m \ll |E|$) and we would like to identify specific links (*i.e.* congested links with large delays) from these measurements. Note that the total delay over a measurement is the sum of delays over the links in a path or connected sub-graph.

Let $x \in \mathcal{R}^{|E|}$ be a non-negative vector whose p-th entry corresponds to the value over link p, and $\mathrm{y} \in \mathcal{R}^m$ denotes the measurement vector whose q-th entry represents the sum of values over the links of q-th measurement which is a connected sub-graph in the network. Let $\mathcal{A}$ be an $m \times |E|$ measurement matrix such that its i-th row corresponds to the i-th measurement. $\mathcal{A}_{ij} = 1$ if and only if the i-th measurement includes link j and zero otherwise. We can write:

$$\mathrm{y}_{m\times 1} = \mathcal{A}_{m\times|E|} x_{|E|\times 1} \tag{1}$$

For example network in Fig. 1 with $|V| = 9$ nodes, $|E| = 10$ links and $m = 3$ measurements, the feasible measurement matrix $\mathcal{A}$ would be:

$$\mathcal{A} = \begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array} \tag{2}$$

The important question is how to estimate the link vector x from the measurement vector y when we have an underdetermined system ($m \ll |E|$). This is possible if we know that the vector x is sufficiently sparse (*e.g.* congested links are much smaller than the set of all links), which is often a reasonable assumption. As mentioned, many features of interest on nodes/links in real-world networks are often sparse. It is worth noting that sparse recovery over networks using compressive sampling has a closely related field called *graph constrained group testing* [3, 10, 22, 35, 39]. They have the same requirements for measurement matrix, but the differences are in: (1) in group testing, x is a logical vector, however, it is a real vector for the CS problem, and (2) the valid operations in each group testing measurement are the logical "AND" and "OR", in contrary to the additive linear mixing of the real numbers in x in CS. [37] showed that compressive sampling can perform better than group testing in terms of the number of required measurements, so we have chosen compressive sampling in this paper.

3 RELATED WORK

There have been a few works which consider graph topological constraints in order to design a feasible measurement matrix in complex networks using compressive sampling [20, 24, 27, 28, 37, 40]. Different methods which were applied for this purpose can be categorized as follows:

- ***Deterministic designs of measurement matrix:*** these methods are based on explicit construction of measurements. In [20], the authors challenged to estimate network link delays from end-to-end probing between boundary nodes along predetermined routes. According to this method, each row of the measurement matrix is constructed with a predetermined path (usually the shortest one) between two boundary nodes. They showed that most of the times with high probability, $0.6|E|$ measurements are sufficient to recover 1-sparse link vectors. Note that this number of measurements is relatively high, regarding to the next methods in the recent literature.
  On the other hand, [37] introduced a new concept called hub. They proposed a method to find a subset of nodes which can be treated as a complete graph and so measured freely via hub. In this method, the additive measurements can be taken over nodes only if they induce a connected sub-graph. They also assume that the constructed measurement matrix is a binary matrix and show that the value of each node in every measurement should be added just once. To achieve this, there must not be more than two nodes with odd degrees in a set of connected nodes by conformity with Euler's theorem. This theorem is not considered in their method, so the constructed measurement matrix is not feasible.
- ***Random designs of measurement matrix:*** in these methods, measurements are generated randomly such that the graph path constraints are held. For example, in [40] each measurement corresponds to a random walk over the links of graph which starts from a random selected node and traverses a number of nodes and links randomly, denoted by t. It is proven that with $t = O(\frac{nD}{c^3 kT(n)})$, the number of required measurements for recovering k-sparse link vectors are $O(c^4 T^2(n) k \log(n))$, where $T(n)$ is the mixing time of random walk and (c, D) are constants. [27, 28] use a different version of random walk which is not biased towards high-degree nodes in order to have a more coverage from the network. In another method [24], each entry of the measurement matrix is randomly generated in nodes and information is propagated through the network using gossip technique which is a decentralized iterative algorithm. In each iteration, a node randomly selects one of its neighbor and their values are updated with the average of their data. As the number of iterations tends to infinity, the data of nodes converge to the average of all of their initial values ($x_i(0)$), denoted by:

$$\lim_{t\to\infty} x_i(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(0) \tag{3}$$

[40] is one of the state-of-the-art CS-based algorithms (we call RW in short) for sparse recovery in networks, although we stated its three major drawbacks in section 1. In this paper, to solve these drawbacks, we propose the CS-SRN framework which reduces the sufficient number of required measurements, increases the accuracy in sparse recovery, and avoids biasedness towards high degree nodes. We will experimentally evaluate the performance of the CS-SRN method compared to RW in section 5.

4 PROPOSED METHOD

In this section, we propose a Compressive Sampling framework for Sparse Recovery in Networks (CS-SRN). In CS-SRN, we construct a feasible random measurement matrix $\mathcal{A}$ to infer the link features inside a network by indirect end-to-end measurements. In the constructed measurement matrix $\mathcal{A}$, each measurement (with non-negative integer entries) has to be feasible in the sense that the links of the same measurement should correspond to a path or connected sub-graph. Thus, CS-SRN holds sparse recovery with network topological constraints. In the proposed method, every row of the measurement matrix $\mathcal{A}$ is constructed from a guided walk based on the CS-SRN approach. In order to construct each of m measurements (row of $\mathcal{A}$), which is called "CS-SRN walk", the following steps should be iteratively performed:

1. A start node is selected relative to the probability $\mathrm{P}(v) = \frac{1}{|V|-1} \times (1 - \frac{deg(v)}{2|E|})$ as current node. This probability has the inverse relation to the degree of a node ($deg(v)$).
2. For all the neighbors of current node, the transition probabilities of moving from current node (Cn) to its neighbor node w where the link $(Cn, w) \in E$ are calculated by $\mathrm{P}(w) = \frac{1}{deg(Cn)} \times \min(1, \frac{deg(Cn)}{deg(w)})$.
3. The probability of staying at current node is computed by $\mathrm{P}(Cn) = 1 - \sum_{w\in\mathcal{N}(Cn)} \mathrm{P}(w)$.
4. The next node is selected between three different options, then the visited link removes. In the first option, if there exist some neighbors for the current node such that their probabilities $\mathrm{P}(w)$ are greater than $\mathrm{P}(Cn)$ then the next node will be chosen randomly among these neighbors. In the second option, if there does not exist such neighbor then the next node is selected relative to $\mathrm{P}(w)$. Otherwise the walker traces back to the previous node in which the next node will be the previously visited node before the current node.

(5) The last three steps are repeated t times which is the length of a walk (measurement).
(6) Finally, we accomplish the above steps m times to construct m independent linear measurements.

Random Walk-based measurement matrix construction of [40] is one of the state-of-the-art CS-based algorithm for sparse recovery in networks. RW-based methods introduces a linear bias towards high-degree nodes [25], and it may be inapplicable for designing an efficient measurement matrix to infer networks with diverse degree distributions, ranging from constant-degree (*e.g.*, in regular graphs), a distribution concentrated around the average value (*e.g.*, in Erdös-Rényi random graphs, or in well-balanced peer-to-peer networks), to heavily right-skewed distributions (as the case in World Wide Web, unstructured P2P, Internet at the IP and Autonomous System level, and Online Social Networks). Because in these networks, the congested links (an example for sparsity property of links) are mostly located on the links pointing to the lower degree nodes.

The basic idea in the proposed CS-SRN framework is inspired by the Metropolis-Hasting MCMC technique [11, 23], which is unbiased toward high-degree nodes [25]. In the proposed method, we can avoid biasedness towards high-degree nodes by selecting a "good start" node for every measurement, and assigning proper probabilities to the neighbors of current node for walking on the best next node for every walk of length t. Moreover, we choose the best next node between three different aforementioned options. Note that these measurements (walks) through the connected sub-graphs show the feasibility of the measurement matrix $\mathcal{A}$. Therefore, the CS-SRN may be more proper approach than RW to solve the aforementioned problem by recovering the congestion links which are located on links pointing to the neighbors of not only high-degree nodes but also low-degree nodes. We experimentally compared the CS-SRN and RW methods under four various aspects and the results in section 5 evidence this claim such that the CS-SRN is more applicable approach than RW for sparse recovery in the networks. As a result, we offer the CS-SRN framework for analysing various kinds of complex networks.

5 EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the CS-SRN framework under various configurations. First, we introduce the synthetic and real-world networks we used for the evaluation. Next, we explain settings of the evaluation. Finally, the achieved results and their analysis are shown.

5.1 Datasets

We consider both synthetic and real-world networks. We use four well-known classic models for generating synthetic networks via SNAP platform [2]. All of these networks have 500 nodes which namely:

(1) The Erdös-Rényi network (The simplest variety of random graphs), [19], with 6000 links.
(2) The Watts-Strogatz network ("Small-world" graphs with high clustering and low path lengths), [38], with 2490 links. The rewiring probability is 0.5 and the number of initial closest neighbors is 5.
(3) The Barabási-Albert network (Graphs with extreme degree distributions, also known as power-law or scale-free graphs), [4], with 2485 links and each new node is preferentially connected to 5 existing nodes.
(4) The $\mathcal{G}^4$ network with each node directly connecting to its four closest neighbors in ring topology with 500 links. $\mathcal{G}^4$ is important to the study of small-world networks [38].

We have summarized these networks in Table 1.

Table 1: Synthetic Networks Models

Synthetic Network Model	Parameters	No. of nodes	No. of links
Erdös-Rényi [19]	-	500	6000
Watts-Strogatz [38]	[5 ; 0.5]	500	2490
Barabási-Albert [4]	5	500	2485
$\mathcal{G}^4$ [38]	4	500	500

We also consider two real-world networks:

(1) The network of 500 busiest commercial airports in the United States (USTop500), [14], with 500 nodes and 2980 links.
(2) The neural network of the Caenorhabditis elegans worm (C.elegans), [38], with 306 nodes and 2345 links.

We have also summarized these networks in Table 2.

Table 2: Real-World Networks

Real-World Network	No. of nodes	No. of links
USTop500 [14]	500	2980
C.elegans [38]	306	2345

5.2 Settings

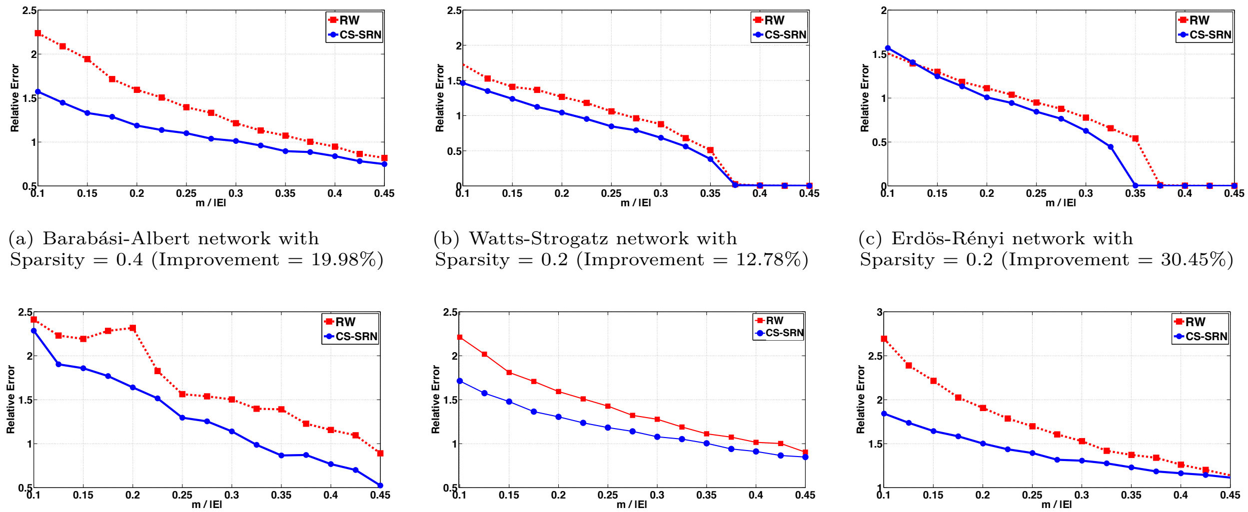
In each of the test cases for the synthetic networks, we generated 3 networks using SNAP with 10 set of walks. For the real-world datasets, we generated 10 set of walks with the same settings. For each network and each set of walks, we performed the experiments. The denoted points in the figures, represent the mean value of all tests.

The objective function that we seek to minimize is the LASSO model [36],[8], which has ℓ_1 norm as the regularization term:

$$\min_x \|x\|_1 + \|\mathcal{A}x - \mathrm{y}\|_2^2. \quad (4)$$

To perform the optimization, we use MATLAB and SPAMS package [1].

In all of the test cases, we compare our CS-SRN method with the work in [40] which we call RW in short. As mentioned before, this work is one of the best existing methods for sparse recovery in networks. For recovery error, we consider the relative error, specifically $\frac{\|x-x'\|_2}{\|x\|_2}$, where x and x' are the original and predicted vectors, respectively.



(a) Barabási-Albert network with Sparsity = 0.4 (Improvement = 19.98%)

(b) Watts-Strogatz network with Sparsity = 0.2 (Improvement = 12.78%)

(c) Erdös-Rényi network with Sparsity = 0.2 (Improvement = 30.45%)

(d) $\mathcal{G}^4$ network with Sparsity = 0.2 (Improvement = 24.7%)

(e) C.elegans network with Sparsity = 0.5 (Improvement = 18.49%)

(f) USTop500 network with Sparsity = 0.8 (Improvement = 16.31%)

Figure 2: Experiment 1: Recovery error for measurements of length $\frac{|E|}{2}$

5.3 Simulations

Experiment 1 (Recovery Error): Figure 2 shows the recovery error for different synthetic and real datasets. The number of measurements (walks) in each dataset is in a range from $0.1|E|$ to $0.45|E|$ and the length of each measurement is set to $\frac{|E|}{2}$. The sparsity percentage for each network is mentioned under its figure. We consider a constant positive value for the non-zero elements and spread them in the unknown link vector x in the following way:

(1) Set N = Sort the nodes by their degrees in ascending order.
(2) Set $l = \phi$ (NULL).
(3) Traverse N in ascending order and for each node, add the connecting links to l.
(4) Set s = number of non-zero elements for the network (from the sparsity percentage).
(5) Set the value of the corresponding elements of the first s links in l to a non-zero constant value.

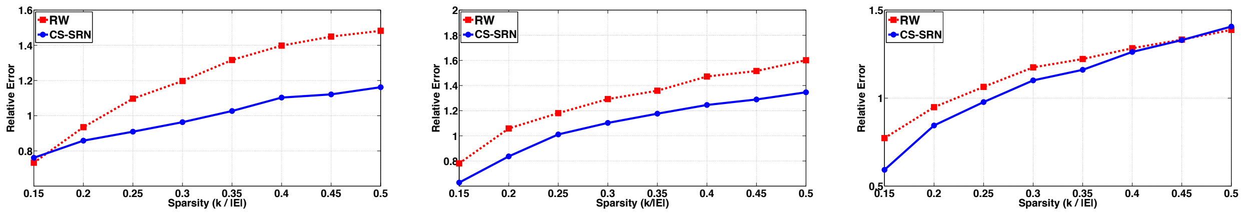
As shown in Figure 2, in almost all test cases, our CS-SRN framework outperforms RW in terms of having lower relative error for all numbers of measurements. Also, our method gets lower error even in small number of measurements compared to RW. This improvement can be very important in the situations where performing measurements has a high cost and the goal is to do an acceptable recovery on a reasonable cost.

The reasons for this improvement in recovery can be explored in several ways. First, in our algorithm, we do not traverse links repeatedly in one measurement due to the options defined in the proposed method, which leads to have a more coverage of the whole network. Second, performing an unbiased CS-SRN walks on the graph, leads to fair coverage of all nodes of all degrees, consequently, the covered links are less connected to high degree nodes, where we are prone to coverage of the links related to such nodes. Hence, in our method, we cover more links and the indirect end-to-end measurements will include more non-zero values by the measurements. Overall, we see around 20.45% improvement in average on all datasets.

Experiment 2 (Sparsity Effect): In this experiment for all networks and for each percentage of sparsity, we ran a set of measurements containing $\frac{|E|}{4}$ walks of length $\frac{|E|}{2}$. The results can be seen in Figure 3. Except a temporal decrease in USTop500 for lower values of k, in all other test cases our method produces a better recovery with lower error for each sparse vector.

In Figure 3, particularly for Watts-Strogatz and Barabási-Albert networks which are closer to real-world graphs than Erdös-Rényi network, it can be observed that even on high sparsity, we have the lower recovery error by our method, and we observe more than 16% improvement in various sparsity. Thus, the results demonstrate that our CS-SRN framework can work accurately even on very high sparse link vectors. Overall, we see around 11.3% improvement in average.

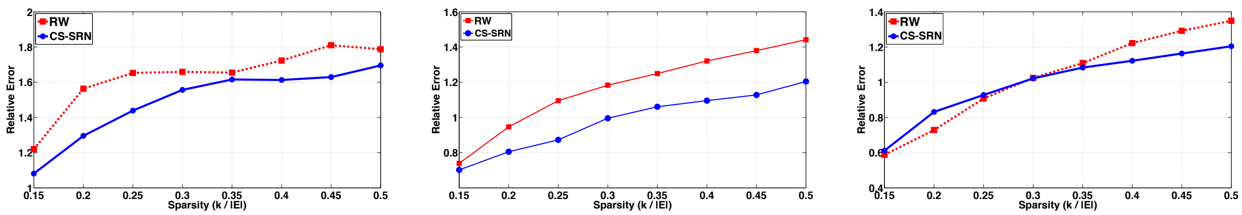
Experiment 3 (Maximum Recoverable Sparsity): In this experiment, with the same settings as experiment 1, we used two variations of Erdös-Rényi networks with 3000 and 6000 links and the $\mathcal{G}^4$ network. For each sparsity for the unknown link vector, we did the recovery of the unknown vectors. We call a vector recovered, if the vector is recovered with



(a) Barabási-Albert network (Improvement = 16.05%)

(b) Watts-Strogatz network (Improvement = 16.15%)

(c) Erdös-Rényi network (Improvement = 6.77%)

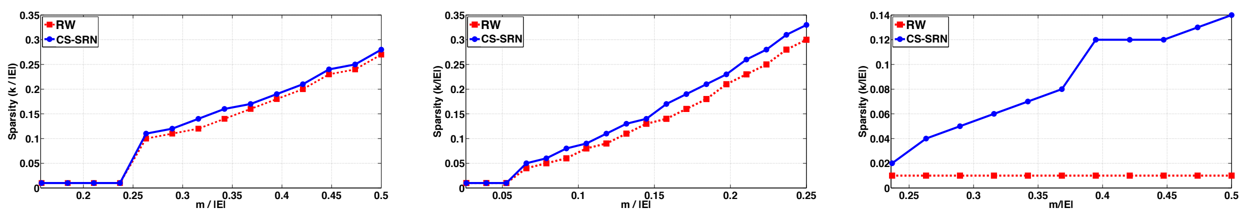


(d) $\mathcal{G}^4$ network (Improvement = 8.94%)

(e) C.elegans network (Improvement = 18.45%)

(f) USTop500 network (Improvement = 1.39%)

Figure 3: Experiment 2: Sparsity effect for $\frac{|E|}{4}$ measurements of length $\frac{|E|}{2}$



(a) Erdös-Rényi network with 3000 links (Improvement = 4.65%)

(b) Erdös-Rényi network with 6000 links (Improvement = 13.95%)

(c) $\mathcal{G}^4$ network (Improvement = 763.64%)

Figure 4: Experiment 3: Maximum recoverable sparsity for measurements of length $\frac{|E|}{2}$

the relative error of no more than 0.01 in at least 90% of generated tests.

As seen in Figure 4, our method gets a better recovery, although this improvement is more significant in $\mathcal{G}^4$ network which has a more structured nature. This can be very useful in the design of the network structure where we can choose the structure that gives us a better recovery of the unknown vector.

Experiment 4 (Impact of Density in Network Structure): In this experiment, we only considered Erdös-Rényi network with different number of links. This causes various density in the graph structure. We consider the unknown link vector to have 20% sparsity and for each network, we ran a set of $\frac{|E|}{2}$ measurements with the length of $\frac{|E|}{4}$ using RW and CS-SRN.

We observe in Figure 5 that in different structure density, the CS-SRN framework has better recovery than RW, while in the density of 0.05, CS-SRN outperforms RW by 33% improvement. It is important to note that in a less dense network structure, the non-zero elements are more spread throughout the network links. Thus, it is natural to have a more difficult recovery in this case. But in a dense network, this task becomes easier and our method uses this privilege in the recovery, while RW does not.

6 CONCLUSION

In this paper, we presented a general framework, called CS-SRN, for the sparse recovery of certain characteristics of the links over the networks in the context of network tomography. Our proposed framework is based on compressive sampling in order to generate a sufficient number of collective additive measurements under network topological constraints to design a feasible measurement matrix. Extensive simulations

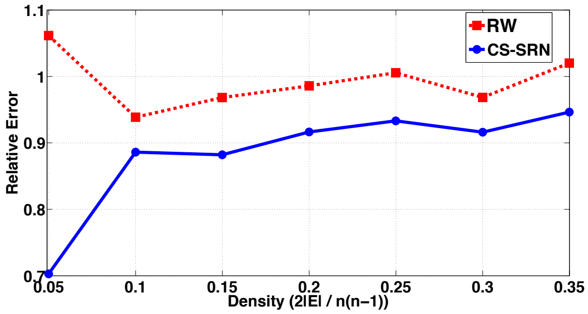


Figure 5: Experiment 4: Impact of Density in Network Structure (Improvement = 10.74%)

have been conducted on both synthetic and real networks in several aspects under various configurations. Our experimental results demonstrate that this framework outperformed the state-of-the-art CS-based method for sparse recovery in a wide class of networks. Further research is needed to find efficient ways to construct measurement paths. Moreover, we have only studied compressive sampling over simple networks, and extensions to weighted networks [28, 30], directed networks, and bipartite networks [33, 34] will be of future works.

REFERENCES

[1] 2017. SPArse Modeling Software (SPAM). (2017). http://spams-devel.gforge.inria.fr/index.html

[2] 2017. Stanford Network Analysis Platform (SNAP). (2017). http://snap.stanford.edu/snap/index.html

[3] P. Babarczi, J. Tapolcai, and P. H. Ho. 2011. Adjacent link failure localization with monitoring trails in all-optical mesh networks. *IEEE/ACM Trans. Netw.* 19, 3 (Jun. 2011), 907–920.

[4] A. L. Barabasi and R. Albert. 1999. Emregence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[5] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. 2008. Combining geometry and combinatorics: a unified approach to sparse signal recovery. In *46th Annual Allerton Conference on Communication, Control, and Computing*. 798–805.

[6] T. Bu, N. Duffield, F.L. Presti, and D. Towsley. 2002. Network tomography on general topologies. In *ACM SIGMETRICS*. 21–30.

[7] E. J. Candes. 2006. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory* 52, 12 (Dec. 2006), 5406–5425.

[8] E. J. Candes, M. Rudelson, T. Tao, and R. Vershynin. 2005. Error correction via linear programming. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 668–681.

[9] E. J. Candes and T. Tao. 2005. Decoding by linear programming. *IEEE Trans. Inf. Theory* 51, 12 (Dec. 2005), 4203–4215.

[10] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama. 2012. Graph constrained group testing. *IEEE Trans. Inf. Theory* 58, 1 (Jan. 2012), 248–262.

[11] S. Chib and E. Greenberg. 1995. Understanding the metropolis-hastings algorithm. *The American Statistician* 49, 4 (1995), 327–335.

[12] A. Coates, A. Hero III, R. Nowak, and B. Yu. 2002. Internet tomography. *IEEE Signal Processing Magazine* 19, 3 (May 2002), 47–65.

[13] M. Coates, Y. Pointurier, and M. Rabbat. 2007. Compressed network monitoring for ip and all-optical networks. In *ACM SIGCOMM IMC*. 241–252.

[14] V. Colizza, R. Pastor-Satorras, and A. Vespignani. 2007. Reaction-diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics* 3 (2007), 276–282.

[15] D. Donoho. 2006. Compressed sensing. *IEEE Trans. Inf. Theory* 52, 4 (Apr. 2006), 1289–1306.

[16] D. Donoho and J. Tanner. 2005. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Natl. Acad. Sci. U.S.A.* 102, 27 (Mar. 2005), 9446–9451.

[17] S.N. Dorogovtsev and J. F. F. Mendes. 2002. Evolution of Networks. *Advances in Physics* 51 (2002), 1079–1187.

[18] N. Duffield. 2006. Network tomography of binary network performance characteristics. *IEEE Trans. Inf. Theory* 52, 12 (Dec. 2006), 5373–5388.

[19] P. Erdos and A. Renyi. 1960. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Science*. 17–61.

[20] M. Firooz and S. Roy. 2011. Link delay estimation via expander graphs. *arxiv:1106.0941* (2011).

[21] S. Ganguli and H. Sompolinsky. 2012. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annual review of neuroscience* 35 (2012), 485–508.

[22] N. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. Chan. 2007. Nonadaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs. In *IEEE INFOCOM*. 697–705.

[23] W. K. Hastings. 1970. Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.

[24] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak. 2008. Compressed sensing for networked data. *IEEE Signal Processing Magazine* 52, 2 (Mar. 2008), 92–101.

[25] M. Kurant, A. Markopoulou, and P. Thiran. 2010. On the bias of BFS. In *International Teletraffic Congress (ITC)*. 1–8.

[26] H. Mahyar. 2015. 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*. 902–909.

[27] H. Mahyar, H. R. Rabiee, and Z. S. Hashemifar. 2013. UCS-NT: An Unbiased Compressive Sensing Framework for Network Tomography. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, Canada*. 4534–4538.

[28] H. Mahyar, H. R. Rabiee, Z. S. Hashemifar, and P. Siyari. 2013. UCS-WN: An Unbiased Compressive Sensing Framework for Weighted Networks. In *Conference on Information Sciences and Systems (CISS), Baltimore, USA*. 1–6.

[29] H. Mahyar, H. R. Rabiee, A. Movaghar, E. Ghalebi, and A. Nazemian. 2015. 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*. 89–96.

[30] H. Mahyar, H. R. Rabiee, A. Movaghar, R. Hasheminezhad, E. Ghalebi, and A. Nazemian. 2015. A Low-cost Sparse Recovery Framework for Weighted Networks under Compressive Sensing. In *IEEE International Conference on Social Computing and Networking (SocialCom), Chengdu, China*. 183–190.

[31] Minh T. N. and K. A. Teague. 2017. Compressive sensing based random walk routing in wireless sensor networks. *Ad Hoc Networks* 54 (2017), 99–110.

[32] S. H. Strogatz. 2001. Exploring Complex Networks. *Nature* 410 (Mar. 2001), 268–276.

[33] S. M. Taheri, H. Mahyar, M. Firouzi, E. Ghalebi K., R. Grosu, and A. Movaghar. 2017. Extracting Implicit Social Relation for Social Recommendation Techniques in User Rating Prediction. In *the 26th International World Wide Web Conference (WWW), Social Computing Workshop: Spatial Social Behavior Analytics on the Web, Perth, Australia*. 1343–1351.

[34] S. M. Taheri, H. Mahyar, M. Firouzi, E. Ghalebi K., R. Grosu, and A. Movaghar. 2017. HellRank: A Hellinger-based Centrality Measure for Bipartite Social Networks. *Social Network Analysis and Mining (SNAM), DOI: 10.1007/s13278-017-0440-7* 7 (May 2017).

[35] J. Tapolcai, B. Wu, P. H. Ho, and L. Rónyai. 2011. A novel approach for failure localization in all-optical mesh networks. *IEEE/ACM Trans. Netw.* 19, 1 (Feb. 2011), 275–285.

[36] R. Tibshirani. 1994. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society B* 58 (1994), 267–288.

[37] M. Wang, W. Xu, E. Mallada, and A.k Tang. 2012. Sparse recovery with graph constraints: Fundamental limits and measurement construction. In *IEEE INFOCOM*. 1871–1879.

[38] D. J. Watts and S. H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684 (1998), 440–442.
[39] B. Wu, P. H. Ho, J. Tapolcai, and X. Jiang. 2010. A novel framework of fast and unambiguous link failure localization via monitoring trails. In *IEEE INFOCOM*. 1–5.
[40] W. Xu, E. Mallada, and A. Tang. 2011. Compressive sensing over graphs. In *IEEE INFOCOM*. 2087–2095.
[41] Y. Zhao, Y. Chen, and D. Bindel. 2009. Towards unbiased end-to-end network diagnosis. *IEEE/ACM Trans. Netw.* 17, 6 (Dec. 2009), 1724–1737.