2015 IEEE International Conference on Smart City/SocialCom/SustainCom together with DataCom 2015 and SC2 2015

# A Low-cost Sparse Recovery Framework for Weighted Networks under Compressive Sensing

Hamidreza Mahyar*, Hamid R. Rabiee†, Ali Movaghar†, Rouzbeh Hasheminezhad*, Elaheh Ghalebi*, Ali Nazemian*
Department of Computer Engineering, Sharif University of Technology (SUT)
*Email: {hmahyar, hasheminezhad, ghalebi, nazemian}@ce.sharif.edu
†Email: {rabiee, movaghar}@sharif.edu

***Abstract*—In this paper, motivated by network inference, we introduce a general framework, called LSR-WEIGHTED, to efficiently recover sparse characteristic of links in weighted networks. The links in many real-world networks are not only binary entities, either present or not, but rather have associated weights that record their strengths relative to one another. Such models are generally described in terms of weighted networks. The LSR-WEIGHTED framework uses a newly emerged paradigm in sparse signal recovery named compressive sensing. We study the problem of recovering sparse link vectors with network topological constraints over weighted networks. We evaluate performance of the proposed framework on real-world networks of various kinds, in comparison with two of the state-of-the-art methods for this problem. Extensive simulation results illustrate that our method outperforms the previous methods in terms of recovery error for different number of measurements with relatively low cost.**

***Index Terms*—Weighted Networks; Low-Cost Sparse Recovery; Compressive Sensing; Social Networks;**

## I. INTRODUCTION

A large number of real-world systems are structured in the form of networks, collections of nodes joined in pairs by links. Typical examples of these networks include large communication systems (*e.g.* Internet, telephone network), technological and transportation infrastructures (*e.g.* railroad and airline routes), biological systems (*e.g.* gene and/or protein interaction networks), information systems (*e.g.* network of citations between academic papers), and a variety of social interaction structures (*e.g.* online social networks) [1; 2]. Social networks, specifically online social networks, have recently emerged as the most popular applications since the Web began in the early 1990s. The popularity of social networks has skyrocketed within the past decades, with the most popular having at present hundreds of millions of registered users. According to Alexa [3], a well-known traffic analytic website, `Facebook` is the second most visited website on the Internet (after Google).

Despite their attractions, extraction of useful knowledge from the network leads to collection and analysis of network data. In recent years, a wide range of researches have explored the social networks for understanding their structure [4; 5], advertising and marketing [6], and others [7]. However, there are two fundamental limitations which make it difficult or impossible to obtain direct measurement of each individual node/link in the network. First, with the growth of technology, social networks are growing rapidly, *i.e.* Facebook has attracted more than 1.4 billion monthly active users worldwide as of March 2015. Second, the global topological structure of many networks is initially unknown, *i.e.* there are access limitations in most social services due to login requirement, API query limits, and so on. However, inferring internal characteristics of networks (nodes/links) from *indirect* end-to-end measurements is relatively inexpensive to obtain. On the other hand, in large-scale networks, it is remarkable to develop methods that can recover high-dimensional unknown node/link characteristics from a small number of measurements. This is still possible if we have a prior knowledge about some properties of links, *i.e.* sparsity. Sparsity is a natural assumption in some social network analysis (SNA) problems such as detection of inter-community links in social networks [8] and identification of top-$k$ central nodes in networks [9]. Because the number of such nodes (or links) are much smaller than the set of all nodes (or links) in social networks. In SNA, estimation of high-dimensional link vectors from a smaller number of measurements is called sparse recovery problem.

More recently, the sparse recovery is achieved through the *compressive sensing theory* which indicates that by taking advantages of the sparsity property, one can efficiently and accurately recover high-dimensional vectors from a much smaller number of measurements or incomplete observations. Compressive Sensing (CS) [10–12] is a recently emerged paradigm in information theory and signal processing for sparse signal recovery which aims to sample and compress sparse signals simultaneously. The main idea behind CS is that the under-sampled data of a signal have all the information needed about that signal, in a proper lower dimensional representation (*i.e.* sparse vector, low-rank matrix, etc.) [10]. The developments in CS began with the seminal works in [12] and [11]. The authors stated that the combination of $\ell_1$-minimization and random matrices can lead to efficient recovery of sparse vectors. They also represented that such concepts have potential to be utilized in wide-ranging applications from astronomy, biology, medicine, to image and video processing. For the last couple of years, CS has been attending in signal processing, but its applications in networking is still in its early stages due to some challenges. One of the most limiting challenges is the construction of measurement matrix that should be feasible with two different constraints:

978-1-5090-1893-2/15 $31.00 © 2015 IEEE
DOI 10.1109/SmartCity.2015.68

IEEE computer society

**(1)** In networks, a measurement matrix is in a more restrictive class taking only non-negative integer entries, while random Gaussian measurement matrices are usually used in the CS literature.

**(2)** More substantially in networks, measurements are restricted by network topological constraints which is not considered in existing CS researches. In other words, only nodes that induce a connected sub-graph can be aggregated together in the same measurement.

There have just been a few recent works that consider network topological constraints in order to construct a feasible measurement matrix over networks (graphs) using compressive sensing [13–18]. All of the previous works (except [18]) for CS applications in networking have not taken into account the strengths of links. These studies rely critically on the assumption that the underlying topology of network is a simple graph and the generation of measurements do not impose any considerable running cost. This assumption is not necessarily true, because in a large number of real-world systems, the heterogeneity in strength, intensity or capacity of connections (links) may be very significant in understanding these networks [19]. On one hand, [20] stated that the strength of social relationships in social networks is a function of their duration, emotional intensity, intimacy, and exchange of services. On the other hand, weights for non-social networks usually refer to the function performed by links, such as the carbon flow between predator-prey pairs in food webs [21], the flux along particular reaction pathways in metabolic biological networks [4], the number of synapses and gap junctions in neural networks [22], or the amount of traffic flowing along connections in transportation networks [19]. Therefore, modelling of such networks should go beyond simple graphs, where links (edges, connections) are not merely binary entities, either present or not, but rather have an associated weight (capacity, intensity, flow, etc.) that record their strengths relative to one another. Such models are generally described in terms of *weighted networks (graphs)* [23].

To the best of our knowledge, the only study for sparse recovery in weighted networks by generating a feasible measurement matrix via compressive sensing is our preliminary work, called UCS-WN [18]. Although, this work improves the state-of-the-art CS-based algorithms for sparse recovery in networks [16], called RW in short, it significantly needs three major improvements:

**(1)** Constructing a feasible measurement matrix with less total number of required measurements for efficient sparse recovery when the relative error should be almost zero.

**(2)** Providing the maximum information coverage using these measurements with a lower total cost (weight).

**(3)** Evaluating weighted networks of various kinds to recover sparse specifications of link vectors with more sparsity and less recovery error.

In this paper, by considering all the aforementioned challenges, we propose a Low-cost Sparse Recovery framework for WEIGHTED networks in an indirect manner by using compressive sensing, called LSR-WEIGHTED. This framework exploits the correlation between the weights and the topological structure of the network, while unveiling the complex architecture of real weighted networks.

The rest of this paper is organized as follows. Section II provides the preliminaries of weighted networks and compressive sensing. The problem addressed in this paper is stated in Section III. Section IV presents the proposed LSR-WEIGHTED framework. The performance evaluation is given in Section V. The concluding remarks are presented in Section VI.

## II. PRELIMINARIES

### A. Weighted Networks

As has long been appreciated, many real-world networks are inherently weighted, in which their links have different strengths. We consider a weighted network, represented by an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes (vertices) with cardinality $|V| = n$, and $E = \{e_1, e_2, ..., e_{|E|}\}$ denotes the set of weighted links (edges) with cardinality $|E|$. Let $Adj$ be the adjacency matrix of $G$. $Adj(u, v) = 1$ if and only if there exists a link between $u$ and $v$, otherwise $Adj(u, v) = 0$. Let $w$ be the weight matrix of $G$, where $w(u, v)$ is the weight of link $(u, v) \in E$.

For a node $v \in V$, we denote its degree by $deg(v)$ and the list of its neighbors by $Nbr(v) \subset V$. A very significant measure of the network properties in terms of the actual weights is obtained by looking at the node strength $s(u)$ as:

$$s(u) = \sum_{v \in Nbr(u)} w(u, v). \tag{1}$$

### B. Compressive Sensing in Networks

Compressive Sensing (CS) is a new theory that has drawn much attention recently for its capability to efficiently acquire and extract sparse information. It has found several applications in various fields such as astronomy, biology, wireless communications, image and video processing, medicine, cognitive radio, and sensor networks [24]. Although CS has been mainly studied in the context of signal and image processing [25], its application in the network inference and analysis is still in the first steps of development.

Let us model and formulate the sparse recovery problem in networks by using the mathematical framework of CS. Consider the network $G = (V, E)$. 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 link vector if $\|x\|_0 = k$, where $\|.\|_0$ is the $\ell_0$-norm which denotes the number of non-zero elements of $x$. In other words, the sparsity of signal $x$ is $k$. For instance, inter-community links have sparsity property in the social networks, so that the number of these links are much smaller than the set of all links in the network [8]. Suppose that we have $m$ measurements over the network which are some connected sub-graph over $G$. Based on the sparse recovery problem (especially compressive sensing) in networks, we would like to efficiently identify these $k$ links from $m$ measurements, by considering the network topological constraints.

Let $x \in \mathcal{R}^{|E|}$ be a non-negative integer vector whose $p$-th entry is the value over link $p$, and $y \in \mathcal{R}^m$ denotes the vector of $m$ measurements whose $q$-th entry represents the total additive values of links in a connected sub-graph over $G$. Let $\mathcal{A}$ be an $m \times |E|$ measurement matrix where its $i$-th row corresponds to the $i$-th measurement. For $i = 1, ..., m$ and $j = 1, ..., |E|$, $\mathcal{A}_{ij} = 1$ if and only if the $i$-th measurement includes link $j$, and zero otherwise. Hence, in the compact form we can write this linear system as:

$$y_{m\times 1} = \mathcal{A}_{m\times |E|}\ x_{|E|\times 1}. \tag{2}$$

In sparse recovery, the set of sparse solutions to this system are of interest and the main question is how to estimate the link vector $x$ from the measurements $y$ in the case of an under-determined system ($m \ll |E|$). In this case, we need to add a constraint that the vectors $x$ are sufficiently sparse to limit the solution space ($k \ll |E|$). Sparsity is often a reasonable assumption in many social network analysis problems. For example, the number of inter-community links are much smaller than the set of all links in the social networks [8] and the number of top-$k$ central nodes are much smaller than the set of all nodes in the network [9].

It is noteworthy that taking measurements on the weighted graphs imposes the running cost of traversed links. The cost of each measurement $i$ is calculated by the sum of the weights of visited links in that measurement, which is denoted by $\mathrm{C}_{m_i}$, and the total cost for recovering sparse vector $x$ would be:

$$\mathrm{C}_{total} = \sum_{i=1}^{m} \mathrm{C}_{m_i}. \tag{3}$$

In this paper, we would like to efficiently recover sparse characteristics of links in social networks from the constructed measurements with a low total cost (weight). We also discuss how to construct a feasible measurement matrix with a lower total links weight regarding to the network topological constraints. Therefore, the constructed measurements via our LSR-WEIGHTED framework for sparse recovery in weighted networks should be feasible and efficient.

## III. Problem Statement

Recently, there has been considerable interest in the analysis of the weighted network model where the social networks are viewed as weighted graphs. Because we can model many real-world systems by weighted networks, considering the heterogeneity in strength or intensity of links along with the complex topological structure. For example, the strength of social relationships is a function of their duration, emotional intensity, intimacy, and exchange of services. The weighted graph model is utilized for analyzing the formation of communities within the network [26], viral and targeted marketing and advertising [6], modeling the structure and dynamics such as opinion formation [27], and for analysis of the network for maximizing the spread of information through the social links [28], in addition to the traditional applications on weighted graphs such as shortest paths, spanning trees, $k$-nearest neighbors, etc. The semantics of the link weights depend on the application (*i.e.* users in a social network assigning weights based on degree of friendship, trustworthiness, behavior, etc.), or the property being modeled (*i.e.* detection of communities [26] or modeling network dynamics [27]).

In social network analysis (SNA), the problem of sparse recovery leads us to address the aforementioned problems. For example, detection of inter-community links is used for community detection [8], identification of high betweenness/closeness centrality nodes is utilized for viral marketing and influence maximization [9], and so on. Therefore, proposing an efficient algorithm to address the problem of sparse recovery in weighted networks is an inevitable task in SNA. In this paper, we want to investigate weighted networks and efficiently recover sparse link characteristics via indirect measurements with relative low total cost (weight) by using the compressive sensing theory.

## IV. Proposed Method

In this section, we propose a Low-cost Sparse Recovery framework for WEIGHTED social networks, called LSR-WEIGHTED, to recover any $k$-sparse link vector of characteristics with relatively low cost. In this framework, we construct a feasible measurement matrix to investigate weighted networks, and efficiently identify the sparse specification of links inside a network via indirect measurements. The constructed measurement matrix from the framework should satisfy the condition of sparse recovery with network topological constraints, in which every measurement with non-negative integer entries has to be feasible in the sense that the links of the same measurement should correspond to a connected sub-graph.

The pseudo code of the proposed framework is shown in Algorithm 1. In this algorithm, every row of the measurement matrix $\mathcal{A}$ is constructed from a measurement based on the LSR-WEIGHTED framework. As depicted, Algorithm 1 includes five steps.

**(1)** A first node ($v_c$) is selected relative to $P_f(v)$ which is calculated for all nodes $v \in V$ based on $Score(v)$, in lines (7)-(11). For each node $v$, $Score(v)$ is defined based on two distinct terms. The first term is relative to the degree of node $v$ and the second term is proportional to the weights of connected links to $v$.

**(2)** The transition matrix is constructed based on the transition probabilities $P_t$ in lines (14)-(19). $P_t(v, u)$ is the probability of moving from node $v$ to node $u$ where $u$ is a neighbor of $v$ and its calculation is relevant to $Score_1(u)$ and $Score_2(u)$. $Score_1(u)$ is a measure based on the degrees of current node $v_c$ and its neighbor $u$. $Score_2(u)$ is a metric related to the weights of links connecting to the current node $v_c$ and the neighbor $u$.

**(3)** The next node is selected proportional to the transition probabilities $P_t(v_c, u)$, in lines (13)-(25). In the algorithm, if there exists no neighbor for the current node $v_c$, the measurement traces back to the recently visited node and continues walking from that node. Then, we hide the traversed link by removing the current node and the next node from the neighbor

**Algorithm 1** The Proposed Framework: LSR-WEIGHTED

**Input:** $G(V, E), m, s$
1: $G(V, E)$: Graph of the weighted network
2: $m$: number of measurements
3: $s$: number of measurement steps
4: $\mathcal{A}$ = NULL /*Initializing Measurement Matrix*/
5: $P_t$ = NULL /*Initializing Transition Matrix*/
6: **for** $i = 1 \to m$ **do** /*Constructing '$m$' measurements*/
7: **Foreach** $v \in V$ **do** /*First Node Selection*/
8: $Score(v) = \frac{deg(v)}{2|E|} + \frac{\sum_{u \in Nbr(v)} w(v,u)}{\sum_{v \in V} \sum_{u \in Nbr(v)} w(v,u)}$
9: $P_f(v) = \frac{1}{n-1}\left(1 - \frac{Score(v)}{\sum_{v \in V} Score(v)}\right)$
10: **end for**
11: $v_c$ = Select first node relative to $P_f(v)$ as current node
12: **for** $j = 1 \to s$ **do**
13: **if** $\exists\, u \in Nbr(v_c)$ **then** /*Next Node Selection*/
14: **Foreach** $u \in Nbr(v_c)$ **do**
15: $Score_1(u) = \frac{1}{deg(v_c)} \times \min\left(1, \frac{deg(v_c)}{deg(u)}\right)$
16: $Score_2(u) = \frac{\sum_{u \in Nbr(v_c)} w(v_c,u)}{w(v_c,u) \times \min\left(1, \frac{\sum_{u \in Nbr(v_c)} w(v_c,u)}{\sum_{z \in Nbr(u)} w(u,z)}\right)}$
17: $Score(u) = Score_1(u) \times Score_2(u)$
18: $P_t(v_c, u) = \frac{Score(u)}{\sum_{u \in Nbr(v_c)} Score(u)}$
19: **end for**
20: $v_n$ = Select next node relative to $P_t(v_c, u)$
21: $Nbr(v_c) = Nbr(v_c) - \{v_n\}$
22: $Nbr(v_n) = Nbr(v_n) - \{v_c\}$
23: **else**
24: $v_n$ = Trace back to the previous node
25: **end if**
26: $v_c = v_n$
27: **end for**
28: Add the measurement to the matrix $\mathcal{A}$ as a new row
29: **end for**

**Output:** feasible measurement matrix $\mathcal{A}$

sets of each other. Updating neighborhoods list enforces the measurement to traverse more links in the network. Thus, the more coverage of the network links yields to have a more accurate solution.

**(4)** The steps (2) and (3) are fulfilled '$s$' times which is the length of a measurement, to generate a new row for the measurement matrix $\mathcal{A}$, in lines (12)-(28).

**(5)** All the previous steps are repeated '$m$' times to construct the measurement matrix $\mathcal{A}$ with '$m$' measurements, in lines (6)-(29).

Now, we describe the key points of the algorithm in details. In LSR-WEIGHTED framework, to efficiently recover sparse link vectors, three different situations for a link in the network $\mathcal{G}$ may happen in order to construct a measurement according to the Algorithm 1: (1) A link is not selected by that measurement, (2) it is visited once by that measurement and then never visited again by that measurement, and (3) it is visited once and if there needs back tracking to the previous node, it is visited for the second time.

Moreover, in the proposed framework, we can avoid biasedness towards high-degree nodes and high-weighted links by selecting a proper start node for measurements, and also assigning suitable probabilities to the neighbors of nodes for selecting the best next node, according to steps (1), (2), and (3). In each measurement, we first select a good start node proportional to the probabilities $P_f$, and then select the best next node relative to the probabilities $P_t$. These probabilities are the combination of nodes degree and links weight. The next node selection step is repeated $s$ times which is the length of a measurement, in step (4). The calculation of transition probability is performed in two phases: Scoring and Normalization. In addition, because of hiding links, it is possible that a node does not have any neighbor to select as a next node, thus, in this case we trace back to the recently visited node. As a result, this kind of measurement over weighted networks leads to have a more and fair coverage of the network links which makes the approach more accurate.

After running the algorithm LSR-WEIGHTED, we have constructed the feasible measurement matrix $\mathcal{A}$ containing $m$ measurements with the step size of $s$ which has non-negative integer entries, as stated in steps (4) and (5). In the proposed framework, each measurement is a connected sub-graph from the network that indicates feasibility of the measurement matrix. Therefore, for detecting the sparse links, our approach satisfies constraint of sparse recovery with the network topological constraints. After generation of measurement matrix $\mathcal{A}$ via the LSR-WEIGHTED framework and adding the accumulative sum of values of visited links to the vector $y$ for each measurement, we form the linear system of Eq. (2). Finally, we want to find the sparse solution for this system, by using the LASSO model as a reconstruction method for the optimization step that is defined by [29; 30]:

$$\min_x \|x\|_1 + \|\mathcal{A}x - y\|_2^2. \tag{4}$$

This framework exploits the correlation between the links weights and the topological structure of the network, unveiling the complex architecture shown by real weighted social networks. We severely offer LSR-WEIGHTED framework for analyzing complex networks, especially social networks. We will experimentally evaluate the performance of our approach with extensive simulations on various networks in the next section.

## V. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the LSR-WEIGHTED framework in social networks of various kinds under several configurations. First, we introduce the real datasets we used for the evaluation. Next, we explain settings of the tests. Then, the achieved results and their analysis are shown. Finally, we analyze the time and space complexity of the algorithm.

### A. Real Datasets

To investigate performance of the proposed method, we consider some well-known real-world weighted social networks with various number of links as test data:

**(1)** The network of personal relationships via Freeman's EIES system (*Freeman*) with 48 nodes and 830 links [31], **(2)** The co-appearance network of characters in the novel Les Miserables (*LesMis*) with 77 nodes and 254 links [32], **(3)** The dynamic contact network of Infectious SocioPatterns (*Infectious*) with 332 nodes and 1781 links [33], **(4)** The social network of American college football games (*Football*) with 35 nodes and 118 links [4], **(5)** The fraternity network of observed and self-reported interactions (*BKFRAT*) with 52 nodes and 3384 links [34].

We also consider some well-known real-world weighted networks to generalize our framework on this complex networks: **(6)** The neural network of the Caenorhabditis Elegans worm (*C.elegans*) with 306 nodes and 2345 links [22], **(7)** The simulation of computer networks (*GRE*) with 1107 nodes and 5664 links [35], **(8)** The chemical engineering plant models for heat exchanger network (*Impcol*) with 207 nodes and 572 links [35], **(9)** The cage model of DNA electrophoresis (*Cage*) with 1015 nodes and 11003 links [36].

### B. Experimental Settings

For each network and each set of measurements, we performed the experiments. In each of the test cases, we generated 10 set of measurements. The denoted points in the figures, represent the median value of the tests for all sets. For recovery error, we consider the relative error as:

$$\frac{\|x - x'\|_2}{\|x\|_2}, \tag{5}$$

where $x$ and $x'$ are the original and estimated vectors, respectively. For the optimization step, we use SPAM package on MATLAB [37]. We choose the LASSO model for the minimization that is described in Eq. (4). In all of the test cases, we compared our LSR-WEIGHTED framework with two recent works: **(1)** The work in [16] which is the state-of-the-art methods for sparse recovery in networks, called RW, and **(2)** The work in [18] for sparse recovery in weighted networks, called UCS-WN, which we mentioned its necessary improvements in section I.

### C. Evaluation Results

**Experiment 1 (Effect of number of measurements on recovery error):** Figure 1 shows the performance evaluation of our framework in comparison with the UCS-WN and RW methods, in terms of recovery error for different number of measurements with relatively low cost. We set the sparsity (the number of non-zero elements) of the unknown vector to 10% of the number of links in each network. The length of each measurement in the test is $\frac{|E|}{5}$. Each point in the horizontal axis is proportional to the number of required measurements divided by the number of all links in the network.

As clearly shown in Table 1, for all networks, our LSR-WEIGHTED framework outperforms UCS-WN and RW in terms of having lower recovery error for different number of measurements. In addition, Figure 1 illustrates that our method gets lower error even in small number of measurements (*i.e.* when the number of measurements is less than 30% of the number of existing links in the network) compared to RW and UCS-WN. 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 with a reasonable cost. The amount of improvement in terms of total cost is also shown in Table 1 for each network. Note that the total cost for constructing the measurement matrix $\mathcal{A}$ via RW, UCS-WN, and LSR-WEIGHTED is calculated based on Eq. (3).

The reason for the better results in recovery error and total cost can be explored in many ways. (1) in our framework we avoid traversing links repeatedly according to the cases defined in the Algorithm 1. This leads to coverage of a greater part of the network with a lower cost, comparing to RW and UCS-WN. (2) an efficient neighbor selection method in the measurements on the network, leads to have a fair coverage of all links with imposing lower weight. Hence, in our approach, we cover more links and the end-to-end measurements will include more non-zero values in each measurement. (3) after each transition we update the lists of neighborhoods to consider changes and have a more accurate solution. Overall, Table 1 shows around 49% (and 32%) and 56% (and 28%) improvements in average on all networks, in terms of recovery error (and total cost), compared to RW and UCS-WN, respectively.

**Experiment 2 (Effect of sparsity percentage on recovery error):** In this experiment for all networks and for each percentage of recovery, we ran a set of measurements containing $\frac{|E|}{5}$ measurements of length $\frac{|E|}{5}$. Figure 2 shows the performance comparison for different sparsity in the unknown vector. It can be observed that even on high sparsity, we have a lower recovery error by our method. As it is shown in Table 2, the LSR-WEIGHTED framework outperforms the RW and UCS-WN methods respectively by around 45% (and 35%) and 45% (and 34%) improvements in average on all networks, in terms of having lower recovery error (and total cost). The reasons for these improvements are the same as experiment 1: (1) The efficient neighbor selection method, (2) More coverage of the network, (3) limiting the traversal of the network links, and (4) Updating the transition matrix in each step of measurements.

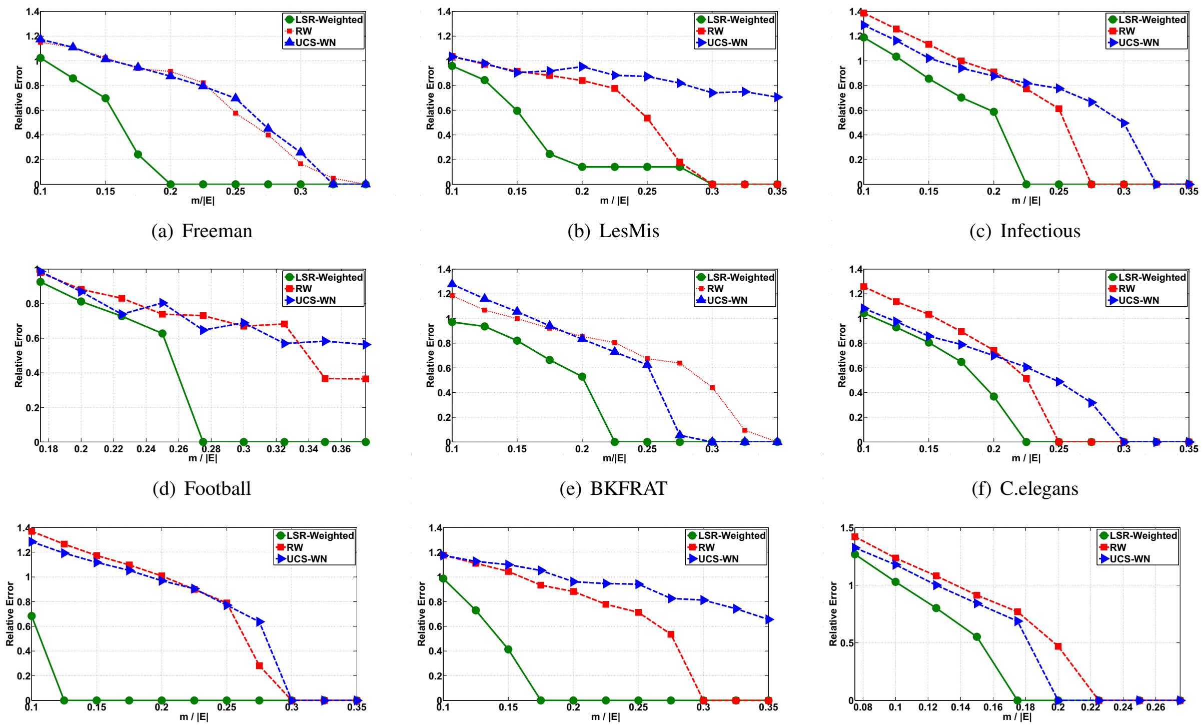
As the final result, it can be seen that LSR-WEIGHTED framework fulfils the requirements that we mentioned for a compressive sensing approach over networks. Therefore, the LSR-WEIGHTED approach is an accurate solution to efficiently recover any $k$-sparse link vector on networks of various kinds even with small number of measurements, high sparsity percentage and relatively low cost.

### D. Complexity Analysis

Consider the network $G = (V, E)$. Assume that each node keeps a hash table data structure for its neighbors, and then checking whether a node is a neighbor of another node can be done in nearly constant time. Sum of the weights of outgoing links from a node like $v \in V$ can be locally computed in $deg(v)$ time and requires $O(1)$ local storage space at each

Table 1: Percentage (%) of improvements comparing to other methods in terms of relative error and total weight (cost) for all networks in Experiment 1

| **Competing** | Freeman | | LesMis | | Infectious | | Football | | BKFRAT | | C.elegans | | GRE | | Impcol | | Cage | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Methods** | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* |
| **RW** | 73.98 | 9.78 | 35.27 | 36.68 | 29.25 | 65.47 | 60.11 | 26.63 | 61.12 | 2.2 | 21.38 | 42.07 | 68.18 | 31.25 | 55.52 | 62.71 | 35.26 | 9.81 |
| **UCS-WN** | 71.23 | 9.05 | 69.46 | 31.73 | 44.86 | 67.09 | 59.58 | 3.02 | 53.32 | 1.91 | 34.49 | 33.6 | 67.89 | 32.97 | 83.05 | 61.89 | 20.87 | 14.09 |




(a) Freeman (b) LesMis (c) Infectious

(d) Football (e) BKFRAT (f) C.elegans

(g) GRE (h) Impcol (i) Cage

Fig. 1: Effect of number of measurements on recovery error for measurement length $\frac{|E|}{5}$ and Sparsity $\frac{|E|}{10}$

node, where $deg(v)$ is the number of elements in the neighbor set of node $v$. Moreover, total sum of the weights of outgoing links for every node can be computed in time $O(n)$, where $|V| = n$. Then, each node can locally compute its score in constant time and sum of the scores of all nodes can be calculated in time $O(n)$. Therefore, the value of $P_f(v)$ is locally computable at the node $v$ in $O(1)$ time, and also selecting the first node relative to $P_f(v)$ function, takes $O(n)$ time. In conclusion, the lines (7)-(11) of the Algorithm 1 can be executed in time $O(n)$ and it requires $O(1)$ local storage space at each node of the network.

Since sum of the weights of outgoing links from a node like $v$ can be locally calculated in $deg(v)$ time using $O(1)$ local storage, each node can locally compute the sum of these computed values for its neighbors in $deg(v)$ time. So the corresponding scores for each node can locally be computed in $O(1)$ time and sum of these scores can be computed in $deg(v)$ time for the node $v$. Then, the transition probabilities for the selected node $v_c$ can be computed in time $deg(v_c)$. Therefore lines (13)-(25) of the Algorithm 1 can be executed in time $O(n)$ using $O(1)$ local storage at each node. Thus, lines (12)-(27) of the algorithm can be executed in time $O\big(n \times s\big)$ using $O(1)$ local storage at each node, where $s$ denotes the measurement length.

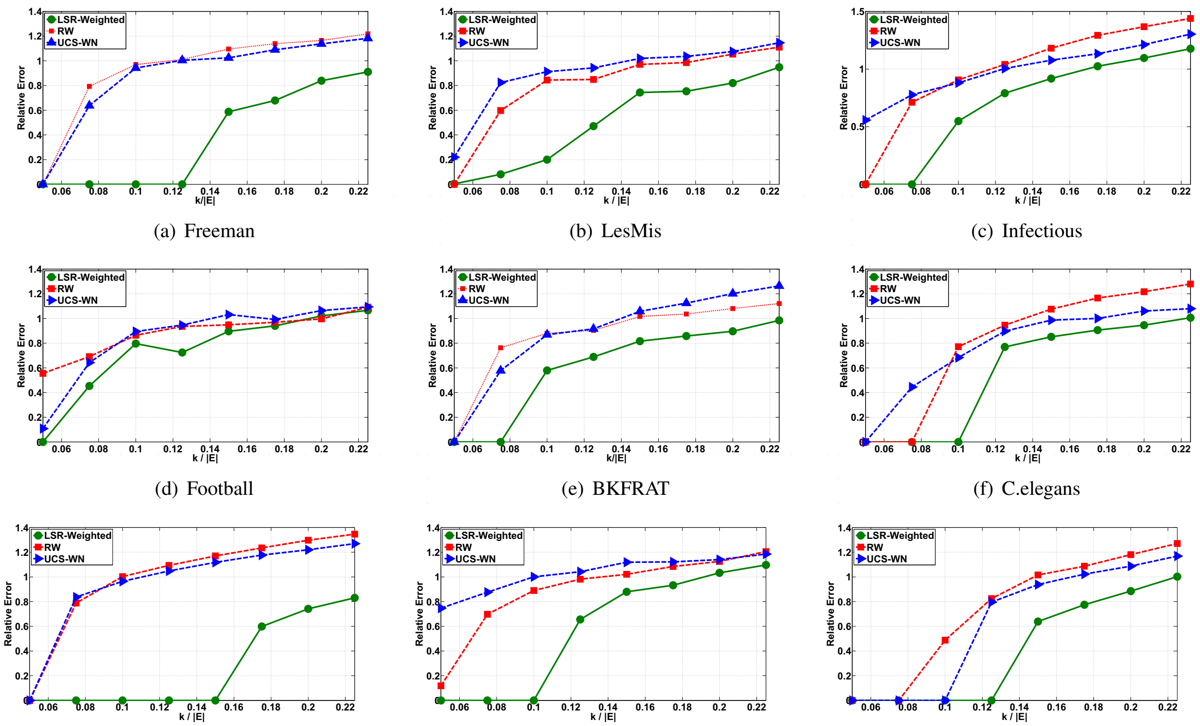
The algorithm inside the outer for loop is executed $m$ times which is equal to the number of measurements. Overall, the final computation time will be $O\big((n+n \times s) \times m\big) = O\big(n \times s \times m\big)$. Furthermore, we will need $O(m \times |E|)$ storage space for the measurement matrix and only require $O(1)$ local storage space at each node.

## VI. Conclusion

In this paper, we addressed the topic of low-cost sparse recovery in weighted networks, in which the links between nodes carry weights representing their heterogeneity in strength, intensity, or capacity. To this end, we proposed a general framework, called LSR-Weighted, in the context of compressive sensing to construct a feasible measurement matrix under network topological constraints. We empirically evaluated the performance of the proposed framework on several real-world networks under various aspects. Simulation results indicated that the LSR-Weighted framework can be employed to efficiently recover sparse link vectors even on low number of measurements, high sparsity, and relatively low cost (weight).

Table 2: Percentage (%) of improvements comparing to other methods in terms of relative error and total weight (cost) for all networks in Experiment 2

| Competing Methods | Freeman | | LesMis | | Infectious | | Football | | BKFRAT | | C.elegans | | GRE | | Impcol | | Cage | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* | *error* | *cost* |
| **RW** | 65.62 | 10.03 | 41.57 | 36.39 | 34.94 | 65.54 | 22.26 | 27.03 | 32.01 | 33.26 | 34.23 | 42.13 | 76.09 | 31.25 | 47.28 | 63.43 | 52 | 9.79 |
| **UCS-WN** | 58.23 | 9.66 | 51.47 | 31.54 | 37.83 | 67.16 | 23.6 | 24.19 | 36.01 | 32.52 | 36.43 | 33.06 | 74.7 | 32.98 | 49.02 | 62 | 37.8 | 14.02 |




(a) Freeman (b) LesMis (c) Infectious

(d) Football (e) BKFRAT (f) C.elegans

(g) GRE (h) Impcol (i) Cage

Fig. 2: Effect of sparsity percentage on recovery error for $\frac{|E|}{5}$ measurements of length $\frac{|E|}{5}$

## References


[1] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, 2001.

[2] S. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," *Advances in Physics*, vol. 51, pp. 1079–1187, 2002.

[3] Alexa Traffic Statistics, July 2015. [Online]. Available: http://www.alexa.com/topsites.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[5] Y. Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *WWW*, 2007.

[6] S. Hill, F. Provost, and C. Volinsky, "Network-based marketing: Identifying likely adopters via consumer networks," *Statistical Science*, vol. 22, pp. 256–275, 2006.

[7] L. Getoor and C. P. Diehl, "Link mining: a survey," *SIGKDD Explor. Newsl.*, vol. 7, no. 2, pp. 3–12, 2005.

[8] H. Mahyar, H. R. Rabiee, A. Movaghar, E. Ghalebi, and 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 2015, Paris, France*, 2015.

[9] 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 2015, PhD Forum, Paris, France*, 2015.

[10] E. J. Cands and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[11] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, 2006.

[12] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[13] M. Firooz and S. Roy, "Network tomography via compressed sensing," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2010, pp. 1–5.

[14] M. Wang, W. Xu, E. Mallada, and A. Tang, "Sparse recovery with graph constraints: Fundamental limits and measurement construction," in *IEEE INFOCOM*, Mar. 2012, pp. 1871–1879.

[15] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 52, no. 2, pp. 92–101, Mar. 2008.
[16] W. Xu, E. Mallada, and A. Tang, "Compressive sensing over graphs," in *IEEE INFOCOM*, Apr. 2011, pp. 2087–2095.
[17] H. Mahyar, H. R. Rabiee, and Z. S. Hashemifar, "UCS-NT: An Unbiased Compressive Sensing Framework for Network Tomography," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013, Vancouver, Canada*, May 2013, pp. 4534–4538.
[18] H. Mahyar, H. R. Rabiee, Z. S. Hashemifar, and P. Siyari, "UCS-WN: An Unbiased Compressive Sensing Framework for Weighted Networks," in *Conference on Information Sciences and Systems, CISS 2013, Baltimore, USA*, Mar. 2013.
[19] A. Barrat, M. Barthlmy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, pp. 3747–3752, 2004.
[20] M. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.
[21] J. J. Luczkowich, S. P. Borgatti, J. C. Johnson, and M. G. Everett, "Defining and measuring trophic role similarity in food webs using regular equivalence," *Journal of Theoretical Biology*, vol. 220, pp. 303–321, 2003.
[22] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[23] S. H. Yook, H. Jeong, A.-L. Barabsi, and Y. Tu, "Weighted evolving networks," *Physical Review Letters*, vol. 86, pp. 5835–5838, 2001.
[24] M. Davenport, M. Duarte, Y. Eldar, and G. Kutyniok, "Introduction to compressed sensing, chapter in compressed sensing: Theory and applications," 2012.
[25] A. C. Sankaranarayanan, P. K. Turaga, R. Chellappa, and R. G. Baraniuk, "Compressive acquisition of dynamic scenes," *CoRR, abs/1201.4895*, pp. 3747–3752, 2012.
[26] J. M. Kumpula, J. P. Onnela, J. Saramaki, K. Kaski, and J. Kertesz, "Emergence of communities in weighted networks," *Physical Review Letters*, vol. 99, 2007.
[27] R. Toivonen, J. M. Kumpula, J. Saramaki, J.-P. Onnela, J. Kertesz, and K. Kaski, "The role of edge weights in social networks: modelling structure and dynamics," *Noise and Stochastic in Complex Systems and Finance*, vol. 6601, pp. B1–B8, 2007.
[28] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003, pp. 137–146.
[29] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, vol. 58, pp. 267–288, 1994.
[30] E. J. Candes, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Oct. 2005, pp. 668–681.
[31] S. C. Freeman and L. C. Freeman, "The networkers network: A study of the impact of a new communications medium on sociometric structure," *Social Science Research Reports*, no. 46, 1979.
[32] D. E. Knuth, "The stanford graphbase: A platform for combinatorial computing," *Addison-Wesley, Reading, MA*, 1993.
[33] SocioPatterns datasets, July 2015. [Online]. Available: http://www.sociopatterns.org/datasets/
[34] K. Romney and S. Weller, "Predicting informant accuracy from patterns of recall among individuals," *Social Networks*, vol. 6, pp. 59–78, 1984.
[35] Sparse Matrix Collection: Harwell-Boeing Group, July 2015. [Online]. Available: http://www.cise.ufl.edu/research/sparse/matrices/HB/index.html
[36] A. Van Heukeluma, G. T. Barkemab, and R. H. Bisselingc, "DNA Electrophoresis Studied with the Cage Model," *Journal of Computational Physics*, vol. 180, 2002.
[37] SPArse Modeling Software (SPAM), July 2015. [Online]. Available: http://spams-devel.gforge.inria.fr/index.html