Kaveh *et al. Appl Netw Sci* (2021) 6:58
https://doi.org/10.1007/s41109-021-00401-7

Applied Network Science

RESEARCH Open Access

Probabilistic network sparsification with ego betweenness

Amin Kaveh*, Matteo Magnani and Christian Rohner

*Correspondence:
amin.kaveh@it.uu.se
InfoLab, Department of Information Technology, Uppsala University, Lägerhyddsvägen 2, House 1, 75105 Uppsala, Sweden

Abstract

Sparsification is the process of decreasing the number of edges in a network while one or more topological properties are preserved. For probabilistic networks, sparsification has only been studied to preserve the expected degree of the nodes. In this work we introduce a sparsification method to preserve ego betweenness. Moreover, we study the effect of backboning and density on the resulting sparsified networks. Our experimental results show that the sparsification of high density networks can be used to efficiently and accurately estimate measures from the original network, with the choice of backboning algorithm only partially affecting the result.

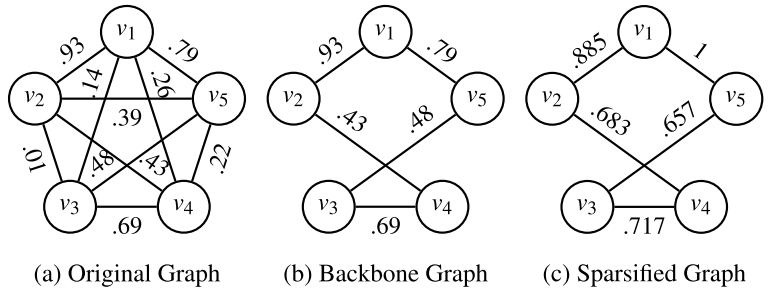
Keywords: Probabilistic network, Ego betweenness, Sparsification, Sampling

Introduction

A probabilistic network is a network in which edges are associated with existence probability. This model has been used in multiple areas such as social networks in which we aim to infer one type of relation (e.g., users' mutual influence) based on another type of interactions (e.g., friendship) (Pfeiffer and Neville 2011) or in road networks in which we require to predict the availability of roads after a catastrophe (Fushimi et al. 2018).

Several studies have been conducted to analyze and evaluate tasks on probabilistic networks such as clustering (Kollios et al. 2011; Han et al. 2019; Ceccarello et al. 2017), core decomposition (Bonchi et al. 2014) and st-reliability (Ke et al. 2019). Most of these studies are based on possible worlds semantics. In possible worlds semantics, all possible instances of a probabilistic network are generated and the intended task is evaluated over all instances. Then the answer to the task will be either a probability distribution (in tasks such as shortest path between two nodes) or a number which is the expectation of the answers to the task over all possible worlds (e.g., source-terminal reliability).

As the number of possible worlds grows by the number of edges exponentially ($2^{|E|}$), following possible worlds semantics is prohibitive even for moderate size networks. To refrain from dealing with such complexity, researchers have chosen two approaches. First, thresholding, in which the edges whose probability is lower than a specific threshold are removed and the remaining edges are kept as deterministic edges. Hence, the resulting network is a deterministic network and there is no need for possible worlds semantics. However, it has been shown that this approach may lead to a considerable



(a) Original Graph (b) Backbone Graph (c) Sparsified Graph

Fig. 1 Procedure of sparsifying a probabilistic graph: *a* input graph to be sparsified, *b* backbone and *c* sparsified graph with our proposed method in which our goal is to preserve nodes' ego betweenness

information loss (De Choudhury et al. 2010). The second approach is sampling in which a percentage of possible worlds are sampled (Jin et al. 2011; Li et al. 2015; Maniu et al. 2017). As probabilistic graphs' entropy[1] is high, the variance of measures over samples and the required number of samples are also high (Parchas et al. 2018; Potamias et al. 2010; Dang et al. 2015; Kahn and Marshall 1953). Therefore, to decrease the required number of samples Parchas et al. proposed the *probabilistic networks sparsification* approach in which the number of edges and graph's entropy are reduced while nodes' expected degrees are preserved by modifying the remaining edges' probabilities (Parchas et al. 2018).

In this work we generalize the definition of *probabilistic network sparsification*. More specifically, we define sparsification as a method to decrease a probabilistic graph's entropy while any specific measure is preserved. This paves the way to examine various topological properties in sparsification. We focus on ego betweenness as a fundamental path-based measure to show the broader applicability of sparsification.

The sparsification procedure includes two steps. (i) In the first step we extract a backbone of the original graph through which we decrease original graph's density[2] and (ii) in the second step we modify the edges' probabilities of the backbone graph. This raises three questions: first, which backboning method leads to a better sparsified graph in terms of preserving the original graph's property? Second, to what extent can we decrease the original graph's density? Third, does the original graph's density have any impact of the quality of the sparsified graph?

Figure 1 shows a probabilistic graph and two steps of sparsification. Figure 1a is a probabilistic graph with 10 edges. Figure 1b shows a backbone extracted from the original graph as the result of the first step. The second column in Table 1 shows the mean relative error (MRE) of expected degree and expected ego betweenness in the backbone. Figure 1c illustrates the second step of sparsification in which edge probabilities are modified such that nodes' ego betweenness are preserved. The third column in Table 1 shows the MRE values of expected degree and expected ego betweenness in this network.

[1] Entropy of network $\mathcal{G}$ is $\mathcal{H}(\mathcal{G}) = \sum_{e \in E} -p_e \log p_e - (1 - p_e) \log (1 - p_e)$ (Parchas et al. 2018).

[2] Density of graph G is $\rho(G) = 2|E|/(|V| \cdot (|V| - 1))$ (Lee et al. 2010).

Table 1 Mean relative error of expected degree and expected ego betweenness in the backbone and the sparsified graph in Fig. 1

	Backbone	Sparsified
MRE—expected degree	0.23	0.10
MRE—expected ego betweenness	0.29	0.17

Contributions

We summarize our contributions as follows:

1. *Generalization:* Sparsification is defined by Parchas et al. as the problem of preserving expected degree. We generalize that definition to the problem of preserving a generic function.
2. *Formalization:* As a specific case of generic function we use expected ego betweenness. Therefore, we formulate how the change of an edge probability alters the ego betweenness of the corresponding nodes.
3. *Evaluating Backbones:* We evaluate the effect of four different backboning algorithms on the resulting sparsified graphs.
4. *Evaluating Graph Density:* We evaluate the effect of the original network density on the resulting sparsified graphs.

The structure of the paper is as follows: "Problem statement" section presents the problem. "Solution framework" section explains the solution framework which includes two algorithms. "Experiments" section includes experimental results and "Discussion and conclusion" section concludes the paper.

Problem statement

A probabilistic network $\mathcal{G} = (V, E, p)$ is a graph (V, E) whose edges are associated with a probability of existence $p : E \rightarrow (0, 1]$.

One of the most practiced approach to analyze probabilistic networks is Monte-Carlo sampling. However, the number of required samples increases as the entropy of the probabilistic graph increases. To decrease the required number of samples, Parchas et al. (2018) have defined probabilistic graph sparsification as:

Definition 1 *Given a probabilistic graph* $\mathcal{G} = (V, E, p)$ *and a sparsification ratio* $0 < \alpha < 1$, *sparsification is to find* $\mathcal{G}^* = (V, E^*, p^*)$, *where* $E^* \subset E$ *and* $|E^*| = \alpha|E|$ *in which* $\sum_{v \in V} |ExD_v - ExD_v^*|$ *and entropy of* G^* *are minimized.*

Where, ExD_v^* is the expected degree of node v in the sparsified graph $\mathcal{G}^*$.

We change this definition to be more inclusive and not to be limited to expected degree. As a result, we can consider other structural properties and examine the effect of changing edge probabilities on that structural property.

Problem 1 *Given a probabilistic network* $\mathcal{G} = (V, E, p)$ *and sparsification ratio* $0 < \alpha < 1$, *sparsification is the process to extract a new probabilistic network* $\mathcal{G}^* = (V, E^*, p^*)$ *where* $E^* \subset E$ *and* $|E^*| = \alpha|E|$, *while* $\sum_{v \in V} |M_v - M_v^*|$ *and entropy of* G^* *are minimized.*

where M and M^* are the values of a structural property in the original graph $\mathcal{G}$ and the sparsified graph $\mathcal{G}^*$ respectively.

In this paper, ego betweenness is the structural property that we aim to preserve. Betweenness of node u is:

$$B(u) = \sum_{s,t \neq u \in V} \frac{g_{st}(u)}{g_{st}} \tag{1}$$

where, $g_{st}(u)$ is the number of shortest paths between s and t passing through u and g_{st} is the total number of shortest paths between s and t (Freeman 1978). In probabilistic networks, the probability of paths decreases as the number of constituent edges increases. Contrary to deterministic networks in which all shortest paths with different length have the same contribution in the calculation of betweenness, lengthy shortest paths have lower effect in the calculation of betweenness in probabilistic networks. Therefore, ego betweenness is a reasonable alternative for betweenness in probabilistic networks. Ego betweenness of node u is the betweenness of u between its immediate neighbors (Everett and Borgatti 2005). Although the calculation of ego betweenness in probabilistic networks is computationally expensive, it can be estimated as:

$$EB_{\mathcal{G}}(u) = \sum_{\{v,w\} \subseteq N(u), v \neq w} p_{uv}\, p_{uw}\, (1 - p_{vw}) \tag{2}$$

where p_{uv} is the probability of the edge between nodes u and v, and $N(u)$ is the set of nodes having an incident edge to u in probabilistic graph $\mathcal{G}$ and if edge $(v, w) \notin E$, we consider $p_{vw} = 0$. Computational complexity is $O(L^2)$ where L is the number of incident edges to node u (Kaveh et al. 2020).

Solution framework

The solution of Problem 1 includes two steps:

1. extract a backbone graph $\mathcal{G}_b = (V, E_b, p_b)$ from the original graph $\mathcal{G} = (V, E, p)$ such that $|E_b| = \alpha|E|$, where α is the sparsification ratio,
2. modify the probability of the edges in E_b such that nodes' ego betweenness is as close as possible to their value in the original graph. The resulting graph is a sparsified graph $\mathcal{G}' = (V, E', p')$, where $|E'| = |E_b| = \alpha|E|$. As we see in "Solution framework" section, $E' = E_b$ in the output of the Gradient-Descent algorithm (see "Gradient-descent (GD)" section) and $E' \subset E$ in the output of the Expectation-Maximization algorithm with E' and E_b not necessarily equal (see "Expectation-maximization (EM)" section).

Therefore, we define ego betweenness discrepancy as follows:

Definition 2 (ego betweenness discrepancy) Given a probabilistic network $\mathcal{G}$ and a sparsified network $\mathcal{G}'$, ego betweenness discrepancy of node u is:

$$\delta(u) = EB_{\mathcal{G}}(u) - EB_{\mathcal{G}'}(u) \tag{3}$$

where, $EB_{\mathcal{G}}(u)$ is the ego betweenness of u in $\mathcal{G}$.

Formally, in the second step of the solution we aim to minimize $D = \sum_{v \in V} |\delta(v)|$. Linear programming (LP) is a possible solution to get the global minimum D. However, it has been shown that not only it is inefficient on large graphs, but also it does not explicitly reduce entropy (Parchas et al. 2018). Therefore, we adapt Gradient-Descent (GD) and Expectation-Maximization (EM) algorithms as in Parchas et al. (2018) to approximate the optimal probability adjustment in a small proportion of time compared to LP while decreasing the entropy. Since in both algorithms we need to have a differentiable function as the objective function, and $\sum_{v \in V} |\delta(v)|$ is not differentiable at 0, then we use $D = \sum_{v \in V} \delta^2(v)$ (Parchas et al. 2018) as the objective function hereinafter.

Backboning

In this section we introduce concisely four backboning methods that have been utilized in this research.

1 *Noise corrected (NC):* The first backboning method is a simplified version of the noise corrected method (Coscia and Neffke 2017; Coscia and Rossi 2019) in which an edge is kept if its probability is higher than the ratio of the sum of the expected degree of its incident nodes divided by the total number of edges connecting to these two nodes. In the NC backboning algorithm the edge under consideration is excluded in the calculation of the ratio.

2 *Maximum Spanning Tree (MST):* The second method is the iterative spanning tree method (Nagamochi and Ibaraki 1992; Parchas et al. 2018). First we construct the backbone graph and initialize it with the same set of nodes in the original graph and empty set of edges. Then, in the first iteration of the algorithm we remove the edges of the spanning tree of $\mathcal{G}$ and add them to the backbone. After that in each iteration we compute the spanning tree/forest of the remaining graph and move the selected edges from remained graph to the backbone. This procedure is repeated until the backbone includes $\alpha|E|$ edges.

3 *Monte Carlo (MC):* The third method is Monte-Carlo sampling through which $\alpha|E|$ edges of the input graph are sampled.

4 *Hybrid (MST/MC):* The forth method is the combination of the second and the third methods (Parchas et al. 2018). First $\alpha'|E|$ edges where $\alpha' < \alpha$ are selected via the iterative spanning tree method and then $(\alpha - \alpha')|E|$ edges are sampled via the Monte-Carlo sampling method.

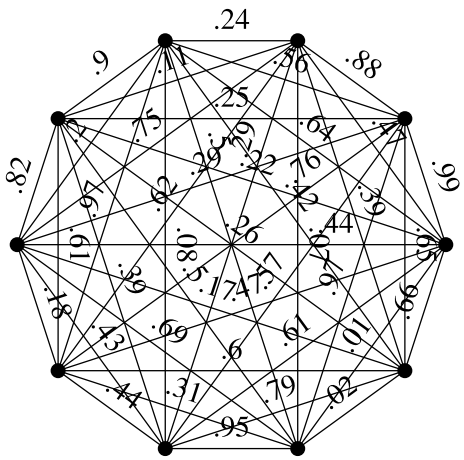


Fig. 2 Probabilistic instance of a complete graph K_{10} with uniformly distributed edge probabilities

We illustrate the differences between the backboning methods on the example of a complete graph K_{10} as shown in Fig. 2. For all backboning methods we repeat the procedures as long as the backbone maintained as single connected component. The first column in Fig. 3 shows the four resulting backbones with $\alpha = 0.31$ resulting from NC, MST, MST/MC ($\alpha' = 0.155$) and MC methods respectively. Although the edges with the highest probabilities are most likely to be represented, there are still considerable differences among the four resulting backbones (e.g., the edge with probability .95 is only present in three of the four backbones.

In the following two sections we describe the *Gradient-Descent* and the *Expectation-Maximization* algorithms where the first one modifies edges' probabilities and the second one rewires backbones as well as modifies edges' probabilities.

Gradient-descent (GD)

Given $\mathcal{G}_b = (V, E_b, p_b)$, the Gradient-Descent algorithm picks one edge $e \in E_b$ in each iteration and optimizes that edge's probability. To achieve this goal, in each iteration we have to reduce the objective function $D = \sum_{v \in V} \delta^2(v)$. According to Eq. 3, if the probability of the edge $e = (u, v)$ changes by ∂p_e^{i+1} at iteration $i + 1$, then the discrepancies of two groups of nodes will change; first the discrepancies of incident nodes to that edge, i.e., u and v and second the discrepancies of common neighbors of the incident nodes. The discrepancies of those nodes that are not members of these two groups do not change because of ∂p_e^{i+1}. Then the derivative of the objective function at iteration $i + 1$ with respect to the change of p_e at that iteration is:

$$\frac{\partial D^{i+1}}{\partial p_e^{i+1}} = -2\frac{\partial \delta^{i+1}(u)}{\partial p_e^{i+1}} - 2\frac{\partial \delta^{i+1}(v)}{\partial p_e^{i+1}} + 2\sum_{w \in W(u,v)} \frac{\partial \delta^{i+1}(w)}{\partial p_e^{i+1}} \tag{4}$$

where $W(u, v)$ is the set of common neighbors of nodes u and v.

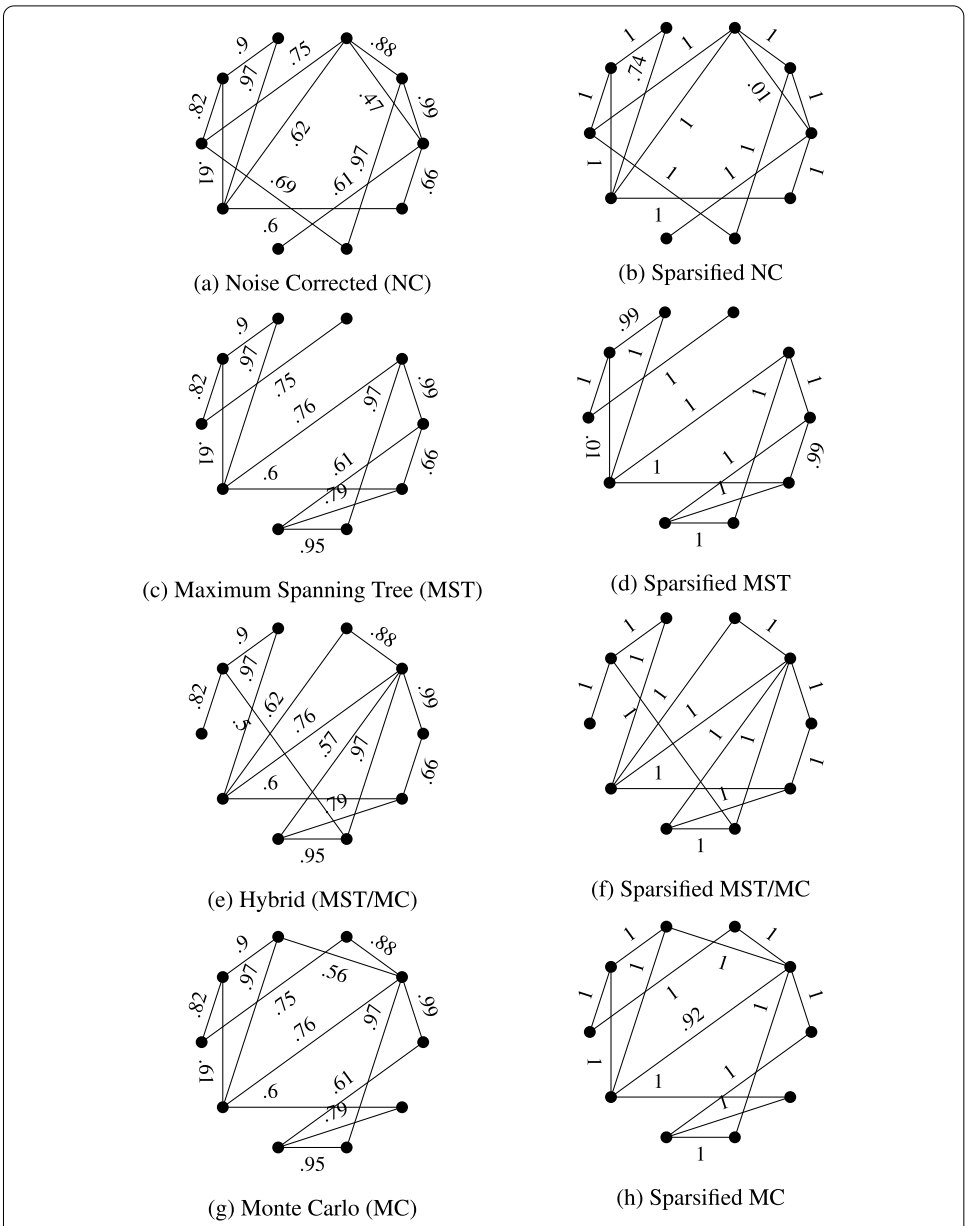


Fig. 3 (First column) Backbones of the original graph in Fig. 2 have been extracted with the methods explained in "Backboning" section (second column) Sparsified graphs of the corresponding backbones resulting from Algorithm 1

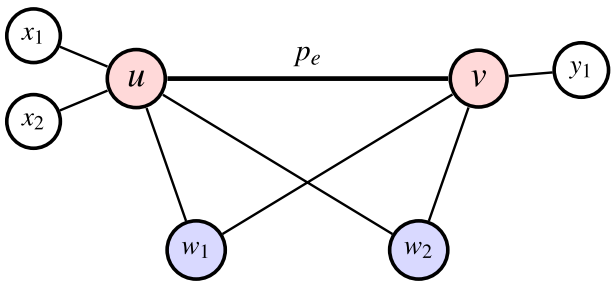


Fig. 4 The effect of changing p_e on discrepancy of other nodes

In Fig. 4, if the probability of edge $e = (u, v)$ increases, the ego betweenness of node u increases, because first nodes x_1 and x_2 rely more on u to be connected to v and second node v relies relatively more on node u to connect to common neighbors w_1 and w_2 if compared to direct connections, i.e., (v, w_1) and (v, w_2). At the same time, if the probability of e increases, the ego betweenness of the common nodes w_1 and w_2 decreases as nodes v and u rely relatively more on their adjacent edge (u, v) in comparison to the two-hop paths that cross nodes w_1 and w_2.

In the following we express the change of discrepancies based on the change of p_e (probability of edge (u, v) in Fig. 4) at iteration $i + 1$ based on the aforementioned intuitions. Equations 5 and 6 represent the change of discrepancies on nodes adjacent to the edge e, and Eq. 7 shows the change of discrepancies of the common neighbors of u and v.

$$\delta^{i+1}(u) = \delta^{i}(u) - (p_{uv}^{i+1} - p_{uv}^{i}) \underbrace{\left(\sum_{w \in W(u,v)} p_{uw}(1 - p_{vw}) + \sum_{x \in N(u) - W(u,v)} p_{ux} \right)}_{C_u} \tag{5}$$

$$\delta^{i+1}(v) = \delta^{i}(v) - (p_{uv}^{i+1} - p_{uv}^{i}) \underbrace{\left(\sum_{w \in W(u,v)} p_{vw}(1 - p_{uw}) + \sum_{y \in N(v) - W(u,v)} p_{vy} \right)}_{C_v} \tag{6}$$

$$\delta^{i+1}(w_j) = \delta^{i}(w_j) - (p_{uv}^{i+1} - p_{uv}^{i}) \underbrace{(-p_{uw_j} p_{vw_j})}_{C(w_j)} \tag{7}$$

where, $N(u)$ are the neighbors of u, and $W(u, v) = N(u) \cap N(v)$ is the set of common neighbors of u and v, and p_{uv}^{i} is the probability of edge (u, v) at iteration i. Hence, by calculating p_{uv}^{i+1} as follows, we will be assured that $\sum_{v \in V} \delta^2(v)$ will get one step closer to the local minimum value:

$$p_{uv}^{i+1} = p_{uv}^{i} + \Delta p \quad , \quad \Delta p = h \frac{\sum_{k \in \{u\} \cup \{v\} \cup W(u,v)} C_k \delta^{i}(k)}{\sum_{k \in \{u\} \cup \{v\} \cup W(u,v)} {C_k}^2} \tag{8}$$

where, $0 < h \leq 1$ is the gradient descent step size. For proof see Appendix 1.

Algorithm 1 illustrates the Gradient-Descent algorithm in which the objective function converges to the local minimum. In line 1, the sparsified graph is initialized with backbone graph ($\mathcal{G}_b$). Then, the algorithm takes iterative steps to reach a local minimum of D. In each iteration, it picks an edge and assigns a new probability to it according to Eq. 8 (line 5). Lines 6-10 assure that the new probability value does not violate constraint $0 \leq p \leq 1$. At the beginning and at the end of each iteration the objective function is calculated in lines 3 and 12 respectively. If the difference between these two values is equal to or lower than the input threshold τ_{GD}, the algorithm finishes.

Algorithm 1: Gradient-Descent Algorithm (GD)

Input: original graph $\mathcal{G} = (V, E, p)$, backbone graph $\mathcal{G}_b = (V, E_b, p_b)$, GD error threshold τ_{GD}

Output: sparsified graph $\mathcal{G}' = (V, E', p')$

1 $\mathcal{G}' \leftarrow \mathcal{G}_b$;
2 **do**
3 $\quad D^i \leftarrow D(\mathcal{G}^i)$;
4 $\quad$ **foreach** $e \in E'$ **do**
5 $\quad\quad p_e^{i+1} \leftarrow p_e^i + \Delta p$;
6 $\quad\quad$ **if** $p_e^{i+1} > 1$ **then**
7 $\quad\quad\quad p_e^{i+1} \leftarrow 1$;
8 $\quad\quad$ **else if** $p_e^{i+1} < 0$ **then**
9 $\quad\quad\quad p_e^{i+1} \leftarrow 0$;
10 $\quad$ **end**
11 $\quad D^{i+1} \leftarrow D(\mathcal{G}^{i+1})$;
12 **until** $|D^{i+1} - D^i| \leq \tau_{GD}$;
13 **return** $\mathcal{G}' = (V, E', p')$;

The second column in Fig. 3 shows the resulting sparsified graphs applied on the four backbones.

Expectation-maximization (EM)

Algorithm 1 only modifies the probability of the edges of the input backbone graph $\mathcal{G}_b$. Therefore, the output sparsified graph $\mathcal{G}'$ is dependent not only on the probability modification of the Gradient-Descent (GD) algorithm but also on $\mathcal{G}_b$. The authors in Parchas et al. (2018) proposed Expectation-Maximization (EM) algorithm that both rewires $\mathcal{G}_b$ and modifies edge probabilities.

The objective function of the EM algorithm is $\sum_{v \in V} \delta^2(v)$. Algorithm 2 illustrates the EM algorithm. The EM algorithm first initializes $\mathcal{G}'$ with the input backbone graph $\mathcal{G}_b$. Lines 2-20, for each edge in E' the algorithm replaces it with an edge in $E \backslash E'$ that yields lower D. In more details, in lines 5-6 the selected edge is removed from $\mathcal{G}'$ and the discrepancies of all corresponding nodes are updated. Line 7, selects the node that has the highest discrepancy, v_c. In lines 8-15, all incident edges to v_c that are not available in the current $\mathcal{G}'$ are examined and the one that has the maximum gain is added to $\mathcal{G}'$. Line 18 runs the GD algorithm to modify edge probabilities of $\mathcal{G}'$ in that iteration. At the beginning and at the end of each iteration the objective function is calculated and if the difference between these two values is equal to or lower than the input threshold τ_{EM}, the algorithm finishes.

The gain of candidate edges are calculated as follow:

$$gain(e_c) = D(\mathcal{G}') - D(\mathcal{G}' + e_c) \tag{9}$$

where, $D(\mathcal{G}')$ is the objective function computed on $\mathcal{G}'$ and $D(\mathcal{G}' + e_c)$ is the objective function computed on $\mathcal{G}'$ after adding e_c.

Algorithm 2: Expectation-Maximization Algorithm (EM)

Input: original graph $\mathcal{G} = (V, E, p)$, backbone graph $\mathcal{G}_b = (V, E_b, p_b)$, EM error threshold τ_{EM}, GD error threshold τ_{GD}

Output: sparsified graph $\mathcal{G}' = (V, E', p')$

$\mathcal{G}' \leftarrow \mathcal{G}_b$;
do
  $D^i \leftarrow D(\mathcal{G}^i)$;
  foreach $e \in E'$ **do**
    $\mathcal{G}' \leftarrow \mathcal{G}' - e$;
    update all corresponding δs;
    $v_c \leftarrow \arg\max\{\delta\}$;
    foreach $e_c \in E \backslash E'$ adjacent to v_c **do**
      $p_c \leftarrow$ according to Equation 8;
      if $p_c > 1$ **then**
        $p_c \leftarrow 1$;
      else if $p_c < 0$ **then**
        $p_c \leftarrow 0$;
      $gain(e_c) \leftarrow$ according to Equation 9;
    end
    $\mathcal{G}' \leftarrow \mathcal{G}' + \arg\max\{gain(e_c)\}$;
  end
  $\mathcal{G}' \leftarrow$ Gradient-Descent Algorithm($\mathcal{G}$, $\mathcal{G}'$,τ_{GD});
  $D^{i+1} \leftarrow D(\mathcal{G}^{i+1})$;
until $|D^{i+1} - D^i| \leq \tau_{EM}$;
return $\mathcal{G}' = (V, E', p')$;

It should be noted that if the probability of an edge becomes zero in the final output of the Gradient-Descent or Expectation-Maximization algorithms, that edge will be removed from the sparsified graph. This is because according to the definition of probabilistic networks, edge probability has to be in the range (0, 1]. As a result, the condition $|E'| = \alpha|E|$ becomes $|E'| \approx \alpha|E|$. Notice that the condition $|E'| = \alpha|E|$ cannot be obtained exactly in practice anyway, because it can define a non-integer number of edges. As a result, the sparsified graph will only contain approximately $\alpha|E|$ edges.

An easy way to avoid probabilities to go to 0 would just be to set a minimum probability of $\epsilon > 0$ in Algorithm 1 line 9. A small ϵ would keep the edge but not have any significant impact on the measures. However, the objective of sparsification is to reduce the size and entropy of the network, so having an algorithm that may lead to a slightly lower size and entropy than requested is practically reasonable in our opinion, without needing any ad hoc fine-tuning.

Experiments

We evaluate the effect of the input graph's density ("Density" section), the impact of the backboning method used ("Backbone" section), and discuss performance ("Performance" section).

Datasets

To evaluate the proposed method, we use three real datasets and six synthetic datasets. While the real datasets give general insights into the scalability and performance on

realistic networks where sparsification will potentially play an important role, we use the synthetic datasets to specifically study the impact of density and network structure.

Brain network

The first dataset is a brain network in which nodes are regions of interest (ROIs). The number of nodes based on the modified version of the standard AAL[3] scheme is 89 (Termenon et al. 2016). This graph is a complete graph and an edge probability is the absolute value of Pearson correlation between the incident ROIs' activity timeseries. A probability value indicates the likelihood that two incident nodes (ROIs) will be functional in the next scanning experiment.

Enron

The second dataset is a snowball sample of the Enron email network in which nodes represent employees and there is an edge between two nodes if at least one email has been exchanged between them. Edge probabilities quantify the likelihood that a new email will be exchanged between a pair of nodes at time t, $p_{i,j} = 1 - \prod_k (1 - exp(-\mu(t - t_k)))$. μ is the scaling parameter, and t_k is the time when message k has been exchanged between nodes i and j (Pfeiffer and Neville 2011).

FriendFeed

A snowball sample of the FriendFeed online social network (Magnani et al. 2010) with 9894 nodes and 172567 edges is the third dataset. There is an edge between two nodes if they follow each other mutually and the probability of that edge is the likelihood that the two incident nodes will exchange a message in the future. This probability is quantified by the exponential function $p_{ij} = 1 - exp(-\mu n)$, where n is the number of messages exchanged between them in any direction and μ is the scaling parameter with the value of 0.2.

Synthetic networks

In addition to the real networks, we also assess multiple synthetic networks, three Erdős–Rényi and three Forest-Fire networks (Leskovec et al. 2005) with densities $\rho = \{0.1, 0.5, 0.9\}$. While Erdős–Rényi is a simple random network, the Forest-Fire network is characterized by a heavy-tailed node degree distribution and community structures. Edge probabilities are assigned using a uniform random distribution between 0 and 1. All datasets are summarized in Table 2.

Density

In this section we study the impact of density and average degree on the sparsification. Figure 5 shows the properties of Erdős–Rényi networks that have been sparsified with the proposed algorithms GD and EM that preserve nodes' ego betweenness (btw). It also shwos GD and EM as proposed in Parchas et al. (2018) that preserve nodes' expected degree (deg). Columns 1-3 represent relative entropy, i.e. $\frac{\mathcal{H}(\mathcal{G}')}{\mathcal{H}(\mathcal{G})}$, mean

[3] Automated Anatomical Labeling is the most widely used anatomical parcellation scheme, which partitions cerebral cortex into 90 parcels (45 for each hemisphere) (Tzourio-Mazoyer et al. 2002).

Table 2 Characteristics of datasets, $|\mathbf{V}|$ is the number of nodes, $|\mathbf{E}|$ is the number of edges, $\overline{\mathbf{p}}$ is the mean of the edge probabilities, $\rho = 2|E|/(|V|(|V|-1))$ is graph density and $\overline{D} = 2|E|/|V|$ is average degree (Lee et al. 2010)

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\overline{p}$	ρ	$\overline{D}$
Brain network	89	3916	0.526	1	88
Enron	805	3956	0.173	0.012	9.83
FriendFeed	9894	172,567	0.11	0.002	34.88
ER (high ρ)	1000	449,550	0.49	0.9	899.1
ER (medium ρ)	1000	249,750	0.5	0.5	499.5
ER (low ρ)	1000	49,950	0.53	0.1	99.9
FF (high ρ)	1000	449,550	0.51	0.9	899.1
FF (medium ρ)	1000	249,750	0.49	0.5	499.5
FF (low ρ)	1000	49,950	0.48	0.1	99.9

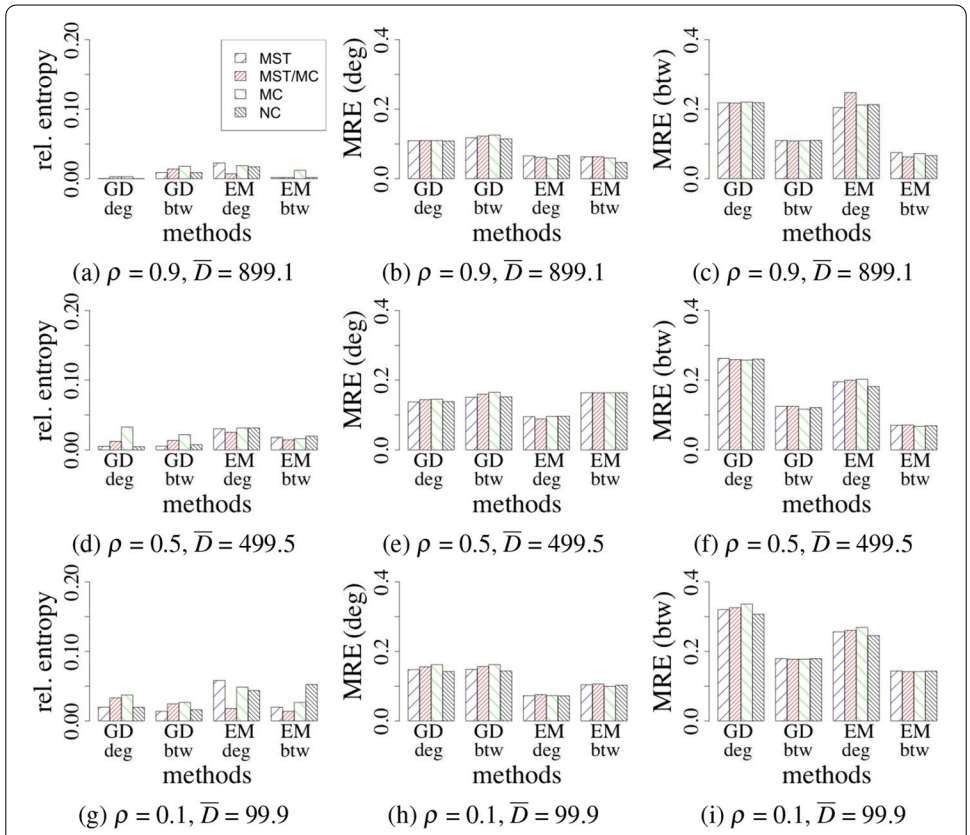


Fig. 5 Impact of density and backboning methods on sparsification - synthetic data: Relative entropy (column one), mean relative error of expected degree (column two) and mean relative error of expected ego betweenness (column three) of sparsified Erdős–Rényi graphs of different density and average degree. The results compare different sparsification methods over various backboning methods. In all graphs sparsification ratio is $\alpha = 0.45$. In MST/MC backboning method $\alpha' = 0.5\alpha$

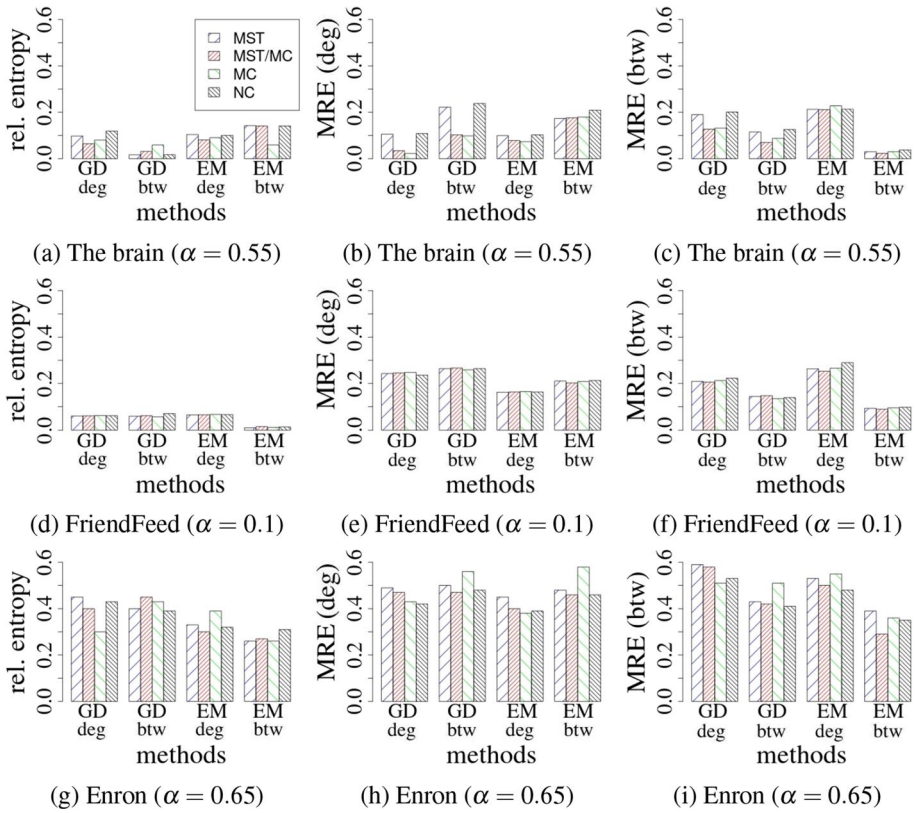


Fig. 6 Impact of density and backboning methods on sparsification–real data: Relative entropy (column one), mean relative error of expected degree (column two) and mean relative error of expected ego betweenness (column three) after sparsification of *a–c* the brain network with $\alpha = 0.55$, *d–f* the FriendFeed network with $\alpha = 0.1$, and *g–i* the Enron network with $\alpha = 0.65$. In the MST/MC backboning method $\alpha' = 0.5\alpha$

relative error (MRE) of expected degree (deg) and mean relative error of ego betweenness (btw) respectively. Rows 1-3 show the results for Erdős–Rényi networks with average degree $\overline{D} = 899.1$, 499.5 and 99.9 accordingly.

Figure 5 shows that all algorithms obtain better results for networks with high density ρ and average degree $\overline{D}$. We have repeated the same experiments over Forest-Fire networks and the results confirm the same conclusion. For the sake of brevity we only include figures for Erdős–Rényi networks. The same evaluations have been performed over the real datasets in Fig. 6. All methods extract low entropy sparsified graphs from the brain network and the FriendFeed network. On the contrary all methods show poor results for the Enron network. This shows that higher average degree of the input graph gives more choices to optimize edge probabilities, and we conclude from these results that sparsification algorithms work better for graphs with high average degree.

Backbone

The structure of the backbone does not seem to have a significant impact on the final sparsified graph. All experiments including Erdős–Rényi (Fig. 5), Forest-Fire as well as

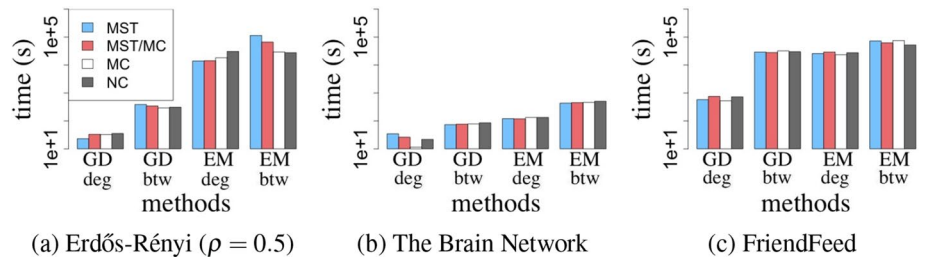


Fig. 7 Time elapsed to sparsify three networks using different sparsification methods. *a* and *b* has been sparsified with $\alpha = 0.45$ and FriendFeed, $\alpha = 0.1$

the real networks (Fig. 6) show only little variation among the backboning algorithms. This happens due to the fact that all backboning methods pick high probability edges as the constituent edges of the backbones (or in the case of Monte-Carlo (MC) method the likelihood that high probability edges are picked is higher). As a result, the majority of the edges in the backbones are common. Our experiments show that between 60 and 80% of the edges in all backboning methods are the same.

Performance

In this section we evaluate the performance of the proposed sparsification methods in the format of computational time (4.4.1), the required number of samples (4.4.2) and precision (4.4.3).

Time complexity

Computational complexity of a node's expected degree and approximated ego betweenness are $O(L)$ and $O(L^2)$ respectively where L is the number of incident edges to that node. Therefore, sparsification based on approximated ego betweenness is computationally more expensive compared to expected degree. Figure 7 shows the time spent to sparsify Erdős–Rényi, the brain and the FriendFeed networks depending on the backboning method. As expected Gradient-Descent (GD) and Expectation-Maximization (EM) with approximated ego betweenness (btw) take more time than GD and EM with expected degree (deg). However, EM (deg) takes more time compared to GD (btw). While time complexity of EM (deg) is higher than GD (btw), Fig. 5 demonstrates that GD (btw) outperforms EM (deg) in reducing entropy. Figure 5c shows that if we use the (btw) algorithms instead of the (deg) algorithms the MRE (btw) error decreases from around 0.2 to around 0.1 while the MRE (deg) error in Fig. 5b increases from around 0.1 to around 0.12. This pattern can be seen in other networks in Figs. 5 and 6.

Number of samples

In this section, we aim to obtain the required number of samples to evaluate measures in sparsified graphs. To estimate the value of measure M, we start with $N_1 = 100$ samples and gradually increase the number of samples until the mean of the measure converges,

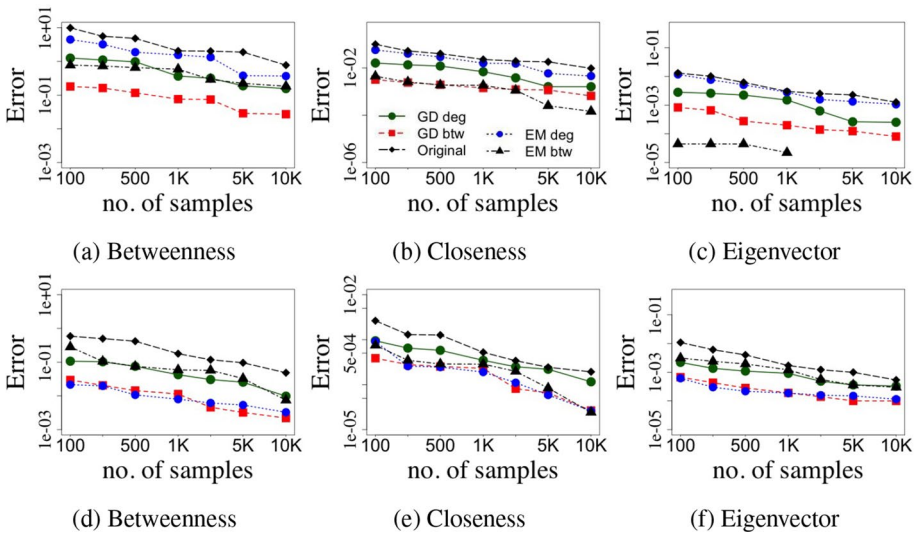


Fig. 8 Required number of samples to calculate measures over sparsified graphs with errors lower than a specific threshold: *a–c* The original graph is Erdős–Rényi with $\rho = 0.1$. The sparsification ratio is $\alpha = 0.45$ and backboning method the maximum spanning tree method, *d–f* the original graph is the brain network, sparsification ratio is $\alpha = 0.55$ and the backboning method is the noise corrected method

i.e. $|\overline{M}_{N_i} - \overline{M}_{N_{i-1}}| < \tau_{error}$, where $\overline{M}_{N_i}$ is the mean value of measure M over N_i samples. In this regard we examine betweenness, normalized closeness (harmonic Marchiori and Latora 2000) and eigenvector centrality. Figure 8 shows that the calculated measures over all sparsified graphs converge with lower number of samples compared to the original graph. This happens because sparsification methods decrease graphs' entropy and subsequently the variance of measures decreases. As a result the required number of samples is lower compared to what is needed for the original graph. Figure 8 shows that measures converge with a lower number of samples if the networks are sparsified with Gradient-Descent (btw). Eigenvector centrality converges faster in the case that the Erdős–Rényi network has been sparsified with the Expectation-Maximization (btw) method. For the brain network all measures converge faster in the sparsified graphs with GD (btw) and EM (deg). However, note that Gradient-Descent (btw) takes less time compared to Expectation-Maximization (deg) for the brain network (see Fig. 7).

Precision

In order to examine precision of measures over sparsified graphs first we compute measures over 50000 samples of the original graph and consider them as the actual value of those measures (although they are estimations of measures, calculating the actual measures are computationally prohibitive). Then, we obtain those measures over 1000 samples of each sparsified graph. Finally, we represent precision of the measures over sparsified graphs by calculating MRE between these two values.

As mentioned in the introduction the majority of measures over probabilistic networks are represented as probability distributions. One of the most fundamental measures is

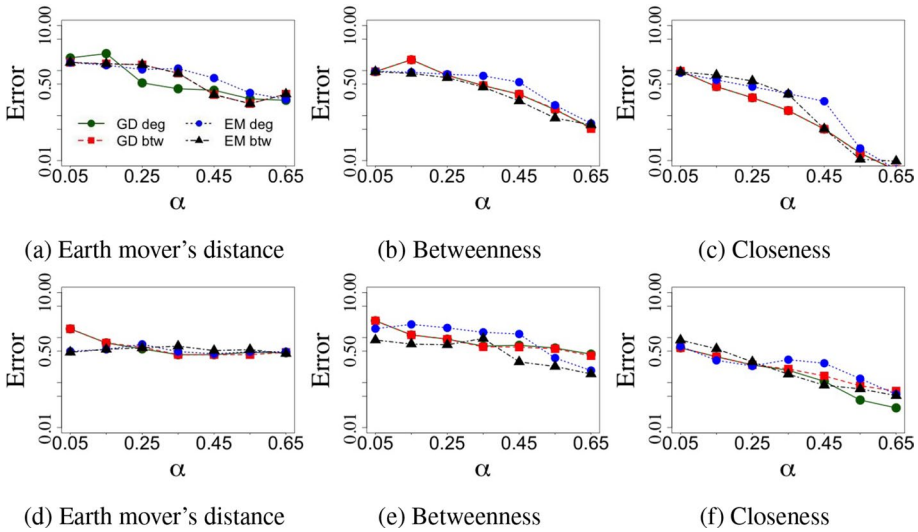


Fig. 9 Error of calculation of measures over sparsified graphs with 1000 samples: In all figures the y-axis shows the difference between the value of measures in the original graphs and the estimated value (over 1000 samples) in the sparsified graphs. The x-axis shows sparsification ratios. *a–c* Erdős–Rényi network with density $\rho = 0.1$and MST/MC backbone ($\alpha' = 0.5\alpha$), and *d–f* the brain network with noise corrected backbone

shortest path length distribution between a pair of nodes. Therefore, instead of comparing the mean value of shortest path length distribution between two nodes we intend to calculate the distance between two distributions. In doing so we require a method to calculate the minimum change that is needed to convert a distribution to another (Parchas et al. 2018). In this regard earth mover's distance (D_{em}) is an appropriate option (Rubner et al. 2000).

Rows 1 and 2 in Fig. 9 represent Erdős–Rényi and the brain networks respectively. The first column in Fig. 9 shows earth mover's distance over various α sparsified graphs. All methods show a similar precision using the earth mover's distance. For small *alpha* the earth mover's distance has lower error if networks are sparsified with Expectation-Maximization (btw) method. Similarly, betweenness can be estimated with lower errors if the networks are sparsified with Gradient-Descent (btw) and Expectation-Maximization (btw) methods (see Fig. 9).

Gradient-Descent (btw) and Expectation-Maximization (btw) are likewise good methods to sparsify networks if we aim to estimate nodes' closeness with a lower number of samples. However, Fig. 9f shows that Gradient-Descent (deg) outperforms other methods for high values of α. This can be explained because of the often high correlation between closeness and degree in networks. As these two measures are highly correlated, sparsifiying a graph while preserving expected degree leads to preserving expected closeness.

Discussion and conclusion

In this paper we generalized the definition of probabilistic network sparsification proposed in Parchas et al. (2018). Our generalized definition is more inclusive and is able to incorporate any topological measure in the process of sparsification. In particular, we examined estimated expected ego betweenness and derived mathematical equations that represent the change of nodes' discrepancies as the function of edge probabilities.

However, we should note that using other topological measures may have scalability issues if we use them naively. A major challenge in probabilistic networks analysis is that all measures are represented in the form of probability distributions and their calculation is expensive. Therefore, developing a closed-formed relation that calculates or estimates each measure may require more research before being able to use it in an efficient sparsification process. Among all measures expected degree can be calculated precisely with $O(L)$ and has been used in sparsification in Parchas et al. (2018) and ego betweenness can be estimated in $O(L^2)$ as done in this paper.

Therefore, using other measures in sparsification algorithms requires (1) calculating/estimating that measure with an efficient time complexity, and (2) finding the relationship of the change of those measures by changing the edges' probabilities.

To evaluate the proposed sparsification methods, we examined various backboning methods (iterative MST, Noise corrected and Monte-Carlo sampling) over multiple synthetic and real datasets. Our experimental results show that the denser a graph is, the better sparsified graphs yield regardless of which sparsification method is used. Better here means lower discrepancies and lower MRE when we compare measures over original and sparsified graphs. This can be explained by the fact that probabilities are real numbers between 0 and 1 and this limits variation of edge probabilities. More precisely we can not increase the values of edge probabilities to be more than 1 in order to compensate positive discrepancies or decrease the values to be less than 0 to compensate negative discrepancies. If the graph is too sparse, the sparsification process may result in all edges having extreme probabilities, which can no longer be updated in the following iterations.

Moreover, it should be noted that the distribution of edge probabilities of the synthetic datasets used in the experiments reported in this paper are uniformly distributed between 0 and 1. We have repeated our experiments for not-skewed (Normal) and skewed (Beta) distributions. While for distributions with a mean lower than 0.5 we have not observed a significant difference with the current experiments, for distributions with higher mean values we have observed lower entropy for the sparsified graphs as well as higher mean relative errors. These errors even increase when we choose small values for α. The intuition on this finding is that when edges probabilities are high on average and we remove $(1-\alpha)$ edges in the backboning stage, discrepancies will be high compared to the case where edges probabilities are low on average. Therefore, Gradient-Descent and Expectation-Maximization algorithms have to minimize discrepancies by adding the probability of the edges available in the sparsified graph. However, as those edges' probabilities are initially high, then

(1) their probability will increase to one and as a result the entropy of the sparsified network will be low (or even zero and in this case the sparsified graph is a deterministic graph) and (2) the minimum discrepancies are still high as edges' probabilities can not be higher than one and as a result the mean relative errors will be high (see Figs. 10 and 11 in Appendix 2). This can affect the estimation of measures such as closeness and shortest path length considerably. For example in the case that the resulting sparsified graph is a deterministic graph and we aim to estimate the reliability between two nodes, it will be estimated either as 0 or 1 which is not a reasonable estimation.

Finally, our experiments show that no sparsification method is consistently outperforming the others. While one method may accurately preserve shortest path length distributions on one network, it does not necessarily have satisfying results for other measures.

Appendix 1: Proof

We notate discrepancy of node v as $\delta(v)$ and define the objective function as $D = \sum_{v \in V} \delta^2(v)$. We know that discrepancy of node v at iteration $i+1$ is:

$$\delta^{i+1}(v) = \delta^i(v) - C_v \Delta p$$

And, the gradient of $\delta^{i+1}(v)$ is:

$$\nabla\delta^{i+1}(v) = \nabla\delta^i(v) - \nabla(C_v \Delta p)$$

Therefore, to minimize the objective function:

$$\begin{aligned}
& \nabla D^{i+1} = \textstyle\sum_{v \in V} 2\delta^{i+1}(v)\nabla\delta^{i+1}(v) = 0 \\
\rightarrow \quad & \nabla D^{i+1} = \textstyle\sum_{v \in V} 2\big(\delta^i(v) - C_v\Delta p\big)\big(\nabla\delta^i(v) - \nabla(C_v\Delta p)\big) = 0 \\
\rightarrow \quad & \nabla D^{i+1} = \textstyle\sum_{v \in V} 2\big(\delta^i(v)\nabla\delta^i(v) - \delta^i(v)\nabla(C_v\Delta p) - C_v\Delta p\nabla\delta^i(v) + C_v\Delta p\big) = 0 \\
\rightarrow \quad & \nabla\Big(\textstyle\sum_{v \in V} \delta^i(v)^2 - \sum_{v \in V} \delta^i(v) C_v \Delta p - \sum_{v \in V} (C_v \Delta p)^2\Big) = 0 \\
\rightarrow \quad & \textstyle\sum_{v \in V} \delta^i(v) C_v \Delta p = \sum_{v \in V} {C_v}^2 \Delta p^2 \\
\rightarrow \quad & \Delta p = \frac{\sum_{v \in V} \delta^i(v) C_v}{\sum_{v \in V} {C_v}^2}
\end{aligned}$$

Appendix 2: Impact of edge probability distributions

To study the impact of edge probability distributions on the sparsification, we compare two groups of distributions, normal and beta (skewed), with means lower and higher than 0.5 on the synthetic datasets. Figures 10 and 11 show relative entropy and MRE errors for Normal and Beta distributions, respectively. The first rows in both figures represent the cases in which the mean of edge probabilities is 0.2. The second rows show the cases in which means of edge probabilities are 0.7. In all cases, the synthetic graphs are sparsified with sparsification ratios $\alpha = 0.1,\ 0.25$, and 0.5. The backboning methods used to sparsify the graphs in Figs. 10 and 11 are NC and MST/MC respectively.

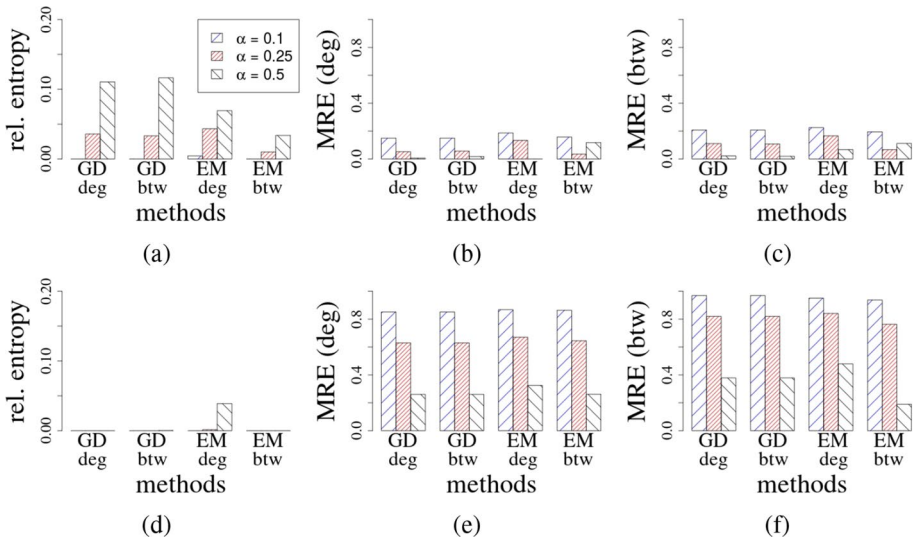


Fig. 10 Normal distributions: relative entropy (column one), mean relative error of expected degree (column two) and mean relative error of expected ego betweenness (column three) of sparsified Erdős–Rényi graphs with 500 nodes and density $\rho = 0.5$. Edge probabilities are assigned according to Normal distributions (row 1) $\mathcal{N}(\mu = 0.2, \sigma = 0.3)$, and (row 2) $\mathcal{N}(\mu = 0.7, \sigma = 0.3)$

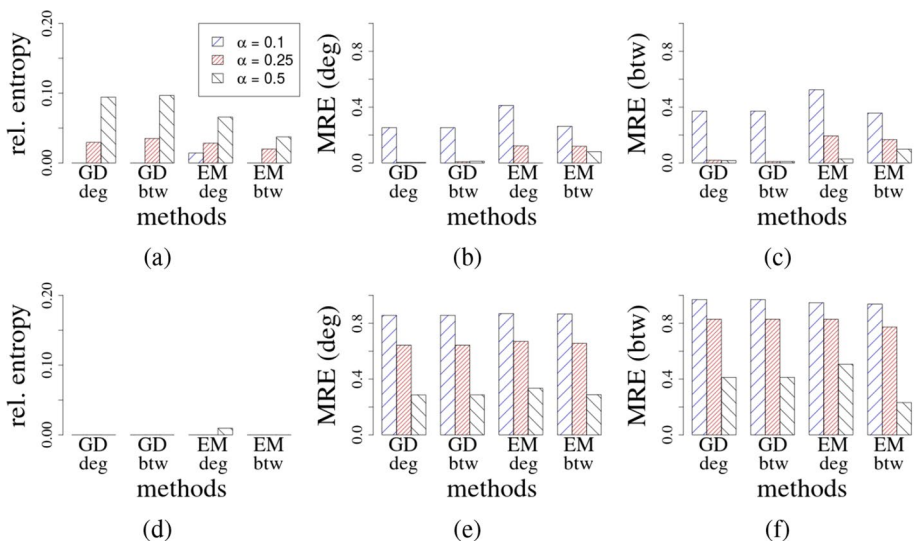


Fig. 11 Beta distributions: relative entropy (column one), mean relative error of expected degree (column two) and mean relative error of expected ego betweenness (column three) of sparsified Erdős–Rényi graphs with 500 nodes and density $\rho = 0.5$. Edge probabilities are assigned according to a Beta distribution with (row 1) $\mathcal{B}(1, 4)$ and mean $= 0.2$, and (row 2) $\mathcal{B}(7, 3)$ and mean $= 0.7$

Abbreviations

ROI: Region of interest; SP: Shortest path; NC: Noise corrected; MST: Maximum spanning tree; MC: Monte-Carlo; GD: Gradient-descent; EM: Expectation-maximization; MRE: Mean relative error.

Acknowledgements

Not applicable

Authors' Contributions
The first author conceived the study, performed the experiments, analysed the data, and wrote the manuscript. M.M. and C.R. analysed the results and wrote the manuscript. All authors approved the final version of the manuscript.

Funding
Open access funding provided by Uppsala University.

Availability of data and materials
The data used during the experimental evaluation of the work is publicly available through the references cited in the paper. The analysis was performed using code developed in R. The codes can be found at https://bitbucket.org/uuinfolab/sparsification-egobtw/

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 17 February 2021 Accepted: 15 July 2021
Published online: 30 July 2021

References

Bonchi F, Gullo F, Kaltenbrunner A, Volkovich Y (2014) Core decomposition of uncertain graphs. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1316–1325
Ceccarello M, Fantozzi C, Pietracaprina A, Pucci G, Vandin F (2017) Clustering uncertain graphs. Proc VLDB Endow 11(4):472–484
Coscia M, Neffke FM (2017) Network backboning with noisy data. In: 2017 IEEE 33rd international conference on data engineering (ICDE). IEEE, pp 425–436
Coscia M, Rossi L (2019) The impact of projection and backboning on network topologies. In: 2019 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 286–293
Dang D-M, Jackson KR, Mohammadi M (2015) Dimension and variance reduction for Monte Carlo methods for high-dimensional models in finance. Appl Math Finance 22(6):522–552
De Choudhury M, Mason WA, Hofman JM, Watts DJ (2010) Inferring relevant social networks from interpersonal communication. In: Proceedings of the 19th international conference on World Wide Web, pp 301–310
Everett M, Borgatti SP (2005) Ego network betweenness. Soc Netw 27(1):31–38
Freeman LC (1978) Centrality in social networks conceptual clarification. Soc Netw 1(3):215–239
Fushimi T, Saito K, Ikeda T, Kazama K (2018) A new group centrality measure for maximizing the connectedness of network under uncertain connectivity. In: International conference on complex networks and their applications. Springer, pp 3–14
Han K, Gui F, Xiao X, Tang J, He Y, Cao Z, Huang H (2019) Efficient and effective algorithms for clustering uncertain graphs. Proc VLDB Endow 12(6):667–680
Jin R, Liu L, Ding B, Wang H (2011) Distance-constraint reachability computation in uncertain graphs. Proc VLDB Endow 4(9):551–562
Kahn H, Marshall AW (1953) Methods of reducing sample size in Monte Carlo computations. J Oper Res Soc Am 1(5):263–278
Kaveh A, Magnani M, Rohner C (2020) Defining and measuring probabilistic ego networks. Soc Netw Anal Min 11(1):1–12
Ke X, Khan A, Quan LLH (2019) An in-depth comparison of st reliability algorithms over uncertain graphs. Proc VLDB Endow 12(8):864–876
Kollios G, Potamias M, Terzi E (2011) Clustering large probabilistic graphs. IEEE Trans Knowl Data Eng 25(2):325–336
Lee VE, Ruan N, Jin R, Aggarwal C (2010) A survey of algorithms for dense subgraph discovery. In: Managing and mining graph data. Springer, Boston, pp 303–336
Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, pp 177–187
Li R-H, Yu JX, Mao R, Jin T (2015) Recursive stratified sampling: a new framework for query evaluation on uncertain graphs. IEEE Trans Knowl Data Eng 28(2):468–482
Magnani M, Montesi D, Rossi L (2010) Friendfeed breaking news: death of a public figure. In: 2010 IEEE second international conference on social computing. IEEE, pp 528–533
Maniu S, Cheng R, Senellart P (2017) An indexing framework for queries on probabilistic graphs. ACM Trans Database Syst 42(2):1–34
Marchiori M, Latora V (2000) Harmony in the small-world. Phys A Stat Mech Appl 285(3–4):539–546
Nagamochi H, Ibaraki T (1992) A linear-time algorithm for finding a sparsek-connected spanning subgraph of ak-connected graph. Algorithmica 7(1–6):583–596
Parchas P, Papailiou N, Papadias D, Bonchi F (2018) Uncertain graph sparsification. IEEE Trans Knowl Data Eng 30(12):2435–2449
Pfeiffer JJ III, Neville J (2011) Methods to determine node centrality and clustering in graphs with uncertain structure. In: ICWSM
Potamias M, Bonchi F, Gionis A, Kollios G (2010) K-nearest neighbors in uncertain graphs. Proc VLDB Endow 3(1–2):997–1008

Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Int J Comput Vis 40(2):99–121
Termenon M, Jaillard A, Delon-Martin C, Achard S (2016) Reliability of graph analysis of resting state FMRI using test-retest dataset from the human connectome project. Neuroimage 142:172–187
Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, Mazoyer B, Joliot M (2002) Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage 15(1):273–289

Publisher's Note