Snowboot: Bootstrap Methods for Network Inference

by Yuzhou Chen, Yulia R. Gel, Vyacheslav Lyubchich and Kusha Nezafati

Abstract Complex networks are used to describe a broad range of disparate social systems and natural phenomena, from power grids to customer segmentation to human brain connectome. Challenges of parametric model specification and validation inspire a search for more data-driven and flexible nonparametric approaches for inference of complex networks. In this paper we discuss methodology and R implementation of two bootstrap procedures on random networks, that is, patchwork bootstrap of Thompson et al. (2016) and Gel et al. (2017) and vertex bootstrap of Snijders and Borgatti (1999). To our knowledge, the new R package **snowboot** is the first implementation of the vertex and patchwork bootstrap inference on networks in R. Our new package is accompanied with a detailed user's manual, and is compatible with the popular R package on network studies **igraph**. We evaluate the patchwork bootstrap and vertex bootstrap with extensive simulation studies and illustrate their utility in application to analysis of real world networks.

Introduction

Traditionally, the structural network analysis, that is, the analysis of data in the form of graphs, has been primarily approached as a descriptive task, in contrast to an inferential task, while the employed analytic tools have rooted mainly within research areas outside of "mainstream" statistical methodology. In the recent years, however, there has been a flare of interest in developing new statistical methods for inference on complex networks (see overviews by Goldenberg et al., 2010; Kolaczyk and Csárdi, 2014; Brugere et al., 2017, and references therein). Despite the explosive growth of statistical methods for network analysis, the developed methodology for inference on graph-structured data remains predominantly parametric and model-based. At the same time, nonparametric bootstrap and resampling appears as an appealing and flexible data-driven alternative for network inference, especially, if only a single realization of a large complex network exists. That is, we can follow the same route as the classical bootstrap of Efron (1979) which was proposed four decades ago as an alternative to conventional parametric methods for independent and identically distributed data, and later was extended to various weakly dependent space-time processes (Hall, 1992; Shao and Tu, 1995; Chernick, 2008).

We attribute the first pioneering attempt to develop nonparametric bootstrap on networks to Snijders and Borgatti (1999). The Snijders–Borgatti procedure, called vertex bootstrap, allows evaluating estimation uncertainty for network density and testing one- and two-sample hypotheses for densities, under the assumption that all the network information is available upfront. Vertex bootstrap is widely used in social studies and is implemented in such highly popular software for social network analysis as UCINET (Borgatti et al., 2002). However, vertex bootstrap remains largely unknown in statistics and there still exists no implementation of this procedure in R. The next attempt to draw bootstrap inference on complex networks, with a focus on uncertainty quantification in network degree estimation, has been suggested by Thompson et al. (2016) and Gel et al. (2017). The idea is to borrow the "blocking" argument, developed for bootstrapping of time series and re-tiling of spatial data, and adapt it to random networks. The resulting procedure, called patchwork bootstrap, starts from sampling a set of multiple ego networks of varying orders and forming a patch (i.e., a network block analogue), and then proceeds to resampling the data within patches. Patchwork bootstrap allows to quantify estimation uncertainties for network degree distribution and its functions, and to construct reliable and sharp confidence intervals in a fully data-driven way. Furthermore, patchwork bootstrap is applicable to ultra-sparse and only partially observable networks.

The new R package **snowboot** (Ramirez Ramirez et al., 2018) is the first to offer vertex and patchwork bootstrap in R. Moreover, to our knowledge, there currently exist only two more R packages implementing bootstrap analysis of networks, **bootnet** and **sna**. The package **bootnet** assesses uncertainty in estimation of edge-weights and centrality indices but does not account for network dependence structure among vertices. The package **sna** implements parametric bootstrapping of edges to generate new random graphs that can be further used for hypothesis testing of matching between the observed network and randomized baseline network as well as canonical correlation coefficients. This paper aims to further fill the gap and showcase utility of nonparametric resampling for network inference, with a particular focus on vertex and patchwork bootstraps. **snowboot** imports the R packages **graphics**, **igraph** (Csárdi and others, 2018), **parallel**, **Rcpp** (Eddelbuettel et al., 2017), **Rdpack** (Boshnakov, 2018), **stats**, and **VGAM** (Yee, 2018).

The paper is organized as follows. In the next section, we discuss methodology and implementation of vertex and patchwork bootstraps. In the simulation studies, we evaluate the implemented bootstrap methods in application to synthetic data. Our case studies illustrate application of the bootstrap to analysis of airline networks, power grids and the David Copperfield network. The paper is concluded by a discussion.

Bootstrap algorithms on networks

Patchwork bootstrap

While the first results on sampling on networks go back to 1960s (see, e.g., Goodman, 1961; Frank, 1968; Granovetter, 1976) and while nowadays there exist numerous graph sampling procedures (see overviews by Scott and Carrington, 2011; Ahmed et al., 2014; Kolaczyk, 2009; Simpson et al., 2015; Zhang et al., 2015, and references therein), still surprisingly little is known on how to reliably and efficiently quantify sampling uncertainties, without imposing typically unverifiable model specification constraints. In this section, we discuss the new method of patchwork sampling and bootstrap (based on algorithms of Thompson et al., 2016 and Gel et al., 2017) that enables us to quantify sampling estimation uncertainties for network degree distribution and its functions, while using only a small proportion of network information.

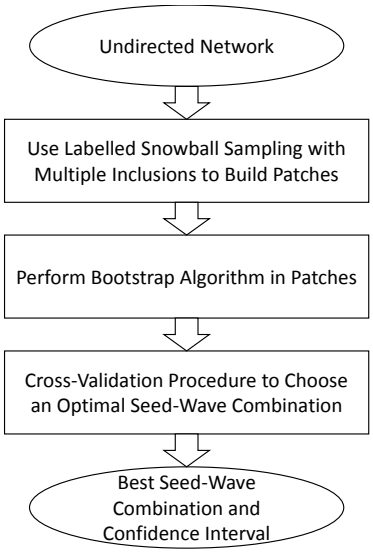

Figure 1: The workflow of the patchwork bootstrap.

Assumptions

Consider an undirected random graph $G = (V, E)$ with a set of vertices, $V(G)$, and a set of edges, $E(G)$. The order and size of G are defined as the number of vertices and edges in G, i.e., $|V(G)|$ and $|E(G)|$, respectively. We assume that G has no self-loops, i.e., $u \neq v$ for any edge $e_{uv} \in E$. The degree of a vertex v is the number of edges incident to v. We denote the probability that a randomly selected vertex has a degree k by $f(k)$, the degree distribution of G by $F = \{f(k), k \geq 0\}$, and the mean degree of G by $\mu(G)$.

Let graph G represent some hypothetical "true" random graph of interest that is never fully observed, and its degree distribution F with finite mean and its order are unknown. Instead, we observe a random graph G_n of order n with a degree distribution $F_n = \{f_n(k), k \geq 0\}$. Let $N_k^{(n)}$ be the number of vertices with a degree k in G_n. Observed graph G_n can be viewed as a realization of G in a sense that as $n \to \infty$, $N_k^{(n)}/n \to f(k)$ in probability (empirical distribution F_n converges in probability to F) and joint degree distribution of G_n approaches that of G (see Britton et al., 2006; van der Hofstad, 2017 and references therein).

Furthermore, we assume that G is *involution invariant* (Lovász, 2012; Orbanz and Roy, 2015; Crane, 2018). Note that involution invariance is linked to unimodularity (Aldous and Lyons, 2007). That is, let us select a vertex v, $v \in V$ and perform a single step random walk from v to one of its neighbors u, $u \in V$; then the distribution of the neighborhood of v will be the same as the distribution of the

neighborhood of u. I.e., from the vantage point of any randomly selected vertex, the rest of the connected network is probabilistically the same. The property of involution invariance can be viewed as a network analogue of stationarity of stochastic processes (Orbanz and Roy, 2015; Crane, 2018). Indeed, as per Aldous and Lyons (2007), unimodularity, or involution invariance, relates to "statistical homogeneity" or "spatial stationarity" of a network. In turn, stationarity is typically an essential condition for consistency of block bootstrap for space and time dependent data, thus, again linking our bootstrap procedure with the "blocking" argument. However, similarly to strong stationarity in time series analysis, involution invariance is not a formally *verifiable* condition. In practical terms of network analysis, the proposed patchwork bootstrap is applicable to networks that are believed to be "homogenous", i.e. their distributional properties are the same across the whole network, which includes, for example, but not limited to exchangeable graphs (Crane, 2018). In turn, the patchwork bootstrap will not be, for example, applicable to networks with community structures. Note that in contrast to exchangeable networks which are dense, involution invariance (and the proposed patchwork bootstrap) is also applicable to sparse networks (Orbanz, 2017).

To the best of our knowledge, there currently exists no competing bootstrap procedure for quantifying uncertainty in estimators of network degree distribution. The subsampling procedure of Bhattacharyya and Bickel (2015) focuses on assessment of uncertainty in motif, or subgraph counts in dense exchangeable networks and is not feasible for inference on degree distribution. The method of Ali et al. (2016) also targets inference on counts of small sub-graphs and is based on the notion of dependency graphs under the network exchangeability framework. The vertex bootstrap approach of Snijders and Borgatti (1999), implemented in our R package **snowboot**, does not impose density limitations but assumes availability of the whole network upfront and is applicable to only small networks. In turn, Lusseau et al. (2008) and Epskamp et al. (2018) propose bootstrap for edge-weights and centrality indices without accounting for network dependence structure among vertices.

The workflow of the proposed patchwork bootstrap consists of the three key steps, that is, Labeled Snowball Sampling with Multiple Inclusions, Resampling, and Cross-Validation (see Figure 1).

Labeled snowball sampling with multiple inclusions

The central element of our technique for network sampling and inference is *patch*, which is a structured sample of vertices and edges joining them. Patch sampling is performed in the following three steps:

1. Sample randomly without replacement several vertices from a network (Figure 2(a)).
2. Construct a Labeled Snowball with Multiple Inclusions (LSMI) independently around each vertex (Figure 2(b)):
   (a) Label each of the sampled vertices as a *seed*.
   (b) Select adjacent vertices by following the edges emanating from the seed and label them as *the first wave of non-seeds*.
   (c) Select and label neighborhoods of second and higher orders by following the edges emanating from the first wave of non-seeds. A vertex with degree $k > 1$ is included into the sample as many times as it is selected by following the previously unused edges (multiple inclusions are allowed).
3. Join all LSMIs into one patch (Figure 2(c)).

Detailed steps of the patch formation are given in Figure 3, which is a modified version of respective algorithms by Thompson et al. (2016) and Gel et al. (2017). The advantage of the algorithm in Figure 3 is that we explore patches with a smaller number of seeds by taking a subset from the seeds we have sampled, rather than by sampling new seeds from a network. This further improves computational efficiency of the algorithm and minimizes the amount of information obtained from a network.

To demonstrate the algorithm, we use one of the artificial networks stored in the R package **snowboot**:

```
> library(snowboot)
> net <- artificial_networks[[1]]
```

Function `lsmi` allows us to create a patch with seeds being randomly sampled or pre-specified. For example, a patch with two random seeds and one wave of neighbors around them:

```
> set.seed(1)
> lsmi(net, n.seed = 2, n.wave = 1)
  [[1]]
  [[1]][[1]]
```

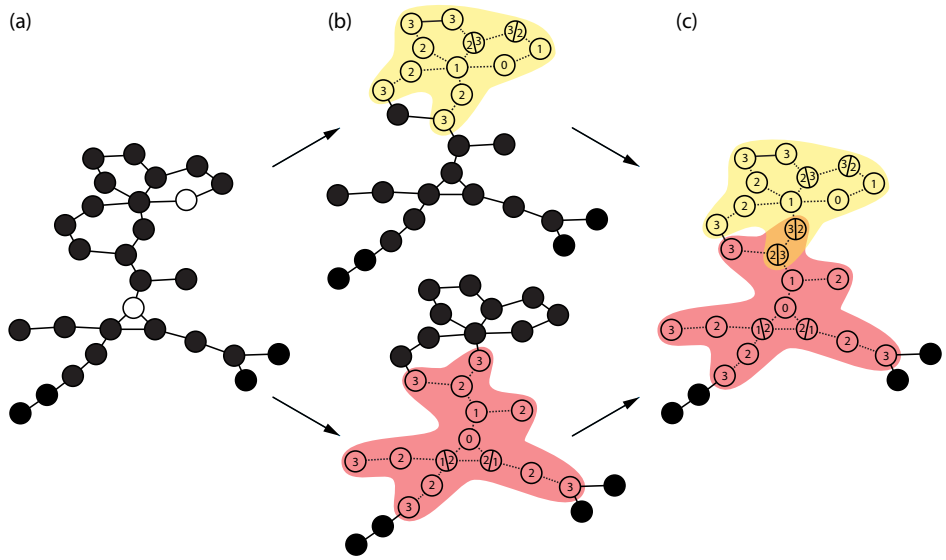

Figure 2: (a) Two seeds sampled from a network; (b) Two LSMIs grown independently, up to the third wave; (c) A patch sample of two seeds and three waves is obtained by joining the two LSMIs together. The waves are denoted with numbers from 0 (seed) to 3 (third wave) in each LSMI.

```
[1] 532
[[1]][[2]]
[1] 524 763

[[2]]
[[2]][[1]]
[1] 744
[[2]][[2]]
[1] 145 858
```

The output is structured as a list of length equal to the number of seeds sampled. Each element of this list is a list itself, where first element contains the seed ID; second element contains IDs of vertices in the first wave, and so on. This structure allows us to keep track of neighborhoods around each seed separately, including the labels – waves in which the vertices appear. The above output shows that two seeds were sampled, with IDs `532` and `744`. The first-order neighbors of vertex `532` are `524` and `763`; of vertex `744` – vertices `145` and `858`.

Alternatively, we can specify particular seeds and select the neighborhoods around them:

```
> lsmi(net, seeds = c(532, 744), n.wave = 1)
```

This option can be used to study specific vertices in a network (e.g., select the neighborhood around Erdös in the Erdös collaboration network of mathematical scientists; Thompson et al., 2016).

The next function, `lsmi_union`, records information about patches obtained by subsetting the originally sampled seeds (Figure 3):

```
> patches <- lsmi_union(net, n.seeds = c(2, 5, 10), n.wave = 2)
> ls(patches)
  [1] "lsmi_big"       "sequence_seeds"
> patches$sequence_seeds
  [[1]]
  [1]  124  352  403  411 1146 1255 1319 1795 1816 1886

  [[2]]
  [1]  411 1146 1319 1816 1886

  [[3]]
  [1] 1146 1886
```

The output is a list of two elements: a patch with the biggest number of seeds and waves from those specified (in the example above, `patches$lsmi_big` is a patch with 10 seeds and 2 waves around each

Input : Graph G_n, where n is number of vertices (network order); numbers of seeds to sample, $\{n.seeds\}_{i=1}^{b}$; number of waves to select, d.

Output: $patch(\max(n.seeds), d) = \{LSMI(d)\}_{i=1}^{\max(n.seeds)}$, i.e., one patch sample for the seed-wave combination with the largest number of seeds and d waves; *sequence_seeds* – a list of length b comprising subsamples of seeds.

`lsmi_union`(G_n, $\{n.seeds\}_{i=1}^{b}$, d)
- $\{n.seeds\}_{i=1}^{b}$ = sort $n.seeds$ in decreasing order
- $\{sequence\_seeds_1\}_{i=1}^{n.seeds_1}$ = sample randomly without replacement $n.seeds_1$ vertices from G_n
- **for** $i = 1, \ldots, n.seeds_1$ **do**
  - $\{LSMI(d)\}_i$ = `lsmi`(G_n, $\{sequence\_seeds_1\}_i$, d)
- **end**
- **for** $j = 2, \ldots, b$ **do**
  - $\{sequence\_seeds_j\}_{i=1}^{n.seeds_j}$ = sample randomly without replacement $n.seeds_j$ vertices from $\{sequence\_seeds_{j-1}\}_{i=1}^{n.seeds_{j-1}}$
- **end**
- **return**

`lsmi`(G_n, *seed*, $n.wave$)
- $\{wave_0\} = seed$
- **for** $i = 1, \ldots, n.wave$ **do**
  - let $\{wave_i\}$ be all immediate neighbors of the vertices from the set $\{wave_{i-1}\}$
  - eliminate all edges that were used to locate $\{wave_i\}$
- **end**
- $LSMI(n.wave) = \{wave_0\} \cup \ldots \cup \{wave_{n.wave}\}$

Figure 3: Obtaining a set of patches with b different number of seeds, and waves from 1 to d.

seed) and a list of seeds and their subsamples (in the example above, those are vectors of lengths 10, 5, and 2 with IDs of the vertices) for constructing smaller patches.

Current version of the LSMI algorithm (Figure 3) pays special attention to counting the edges, thus, to precise recording of the vertices' degrees. In each patch, estimates of the probabilities for a vertex to have a certain degree k ($k > 0$) are obtained with a modified Horvitz–Thompson estimator, whereas $\hat{f}(0)$, the probability of vertices with zero degree, is approximated by the proportion of seeds with zero degree, $\hat{p}_0$. These estimates can be employed to calculate functions of the network degree distribution, e.g., mean degree μ (Thompson et al., 2016; Gel et al., 2017):

$$\begin{aligned} \hat{f}(0) &= \hat{p}_0 = \frac{|\{d_s = 0\}|}{|\{d_s\}|}, \\ \hat{f}(k) &= \frac{|\{d_s = k\}| + (1 - \hat{p}_0)\hat{\mu}_s |\{d_{ns} = k\}| k^{-1}}{|\{d_s\}| + \hat{\mu}_s \sum_{k \geqslant 1} |\{d_{ns} = k\}| k^{-1}}, \\ \hat{\mu}(G) &= \sum_{k \geqslant 0} k \hat{f}(k), \end{aligned} \tag{1}$$

where d_s are the degrees of the sampled seeds; d_{ns} are the degrees of non-seeds; $|\cdot|$ denotes cardinality of a set; $\hat{\mu}_s$ is the estimated mean degree based on $\{d_s\}$:

$$\hat{\mu}_s = \sum_{k \geqslant 0} k \frac{|\{d_s = k\}|}{|\{d_s\}|}.$$

Since the probability of non-seed vertices to be included into an LSMI is proportional to their degree, the estimates $\hat{f}(k)$ in (1) use information from non-seeds downweighted by k^{-1}.

Using the `lsmi_dd` function, calculate $\hat{f}(k)$ from the patch we obtained earlier:

```
> empdd <- lsmi_dd(patches$lsmi_big, net)
> empdd$mu
  [1] 2.40574
> empdd$fk
            0           1           2           3           4           5
  0.000000000 0.369724254 0.248171075 0.207653348 0.082301632 0.040517727
            6           7           8           9
```

```
0.004220597 0.025323579 0.016460326 0.005627462
```

The output is an object of class `'snowboot'` containing the estimates of mean degree and degree distribution. The length of the latter vector depends on what was the maximal degree ($\max(k)$) of the vertices included into the current patch. In the example above, $\max(k) = 9$.

Resampling procedure

To quantify uncertainty associated with the sample estimates (1) of network statistics, we apply a weighted bootstrap procedure. The probability of each non-seed vertex to appear in a bootstrap sample is assigned a weight of d_{ns}^{-1}, i.e., reciprocal of the vertex's degree. We obtain B bootstrap samples and compute network statistic on each of them to approximate the distribution of the sample estimates. Each combination of number of seeds and waves gives different estimates (1), hence, the bootstrap procedure is applied separately to patches of different seed-wave combinations. The bootstrap counterparts of the estimates (1) are as follows:

$$\begin{aligned} \hat{f}^*(0) &= \hat{p}_0^* = \frac{|\{d_s^* = 0\}|}{|\{d_s^*\}|}, \\ \hat{f}^*(k) &= \frac{|\{d_s^* = k\}| + (1 - \hat{p}_0^*)|\{d_{ns}^* = k\}|}{|\{d_s^*\}| + |\{d_{ns}^*\}|}, \\ \hat{\mu}^*(G) &= \sum_{k \geqslant 0} k\hat{f}^*(k), \end{aligned} \tag{2}$$

where $\{d_s^*\}$ and $\{d_{ns}^*\}$ are the respective sets of bootstrapped seeds and non-seeds.

The empirical bootstrap degree distribution can be obtained using the `boot_dd` function from the **snowboot** package:

```
> B <- 50
> bootdd <- boot_dd(empdd, B)
```

A part of the output is a $(1 + \max(k)) \times B$ matrix of bootstrap estimates $\hat{f}^*(k)$, where $k = 0, 1, \ldots, \max(k)$, and a list of length B with $\hat{\mu}^*$:

```
> dim(bootdd$fkb)
  [1] 10 50
> length(bootdd$mub)
  [1] 50
```

The bootstrap degree distributions are used to quantify estimation uncertainty. That is, let η be the parameter of interest (i.e., the population value, μ or $f(k)$), $\hat{\eta}_n^j$ and $\hat{\eta}_n^{j*}$ be the respective conventional and bootstrap estimators of η based on a graph G_n ($j = 1, \ldots, J$ are the different seed-wave combinations for constructing patches). Then the Efron's $100(1-\alpha)\%$ bootstrap confidence interval for η:

$$CI^j_{percentile} = \left(\hat{\eta}^{j*}_{n,[B\alpha/2]},\ \hat{\eta}^{j*}_{n,[B(1-\alpha/2)]}\right), \tag{3}$$

where B is the number of bootstrap samples; $\hat{\eta}^{j*}_{n,[B\alpha/2]}$ and $\hat{\eta}^{j*}_{n,[B(1-\alpha/2)]}$ are the empirical quantiles from the bootstrap distribution.

Having all bootstrapped values available from the output of `boot_dd`, we can employ a variety of methods for calculating bootstrap intervals alternative to the percentile method (3). At this time, the function `boot_ci` can be switched to compute "basic" bootstrap intervals (see Equation 5.6 by Davison and Hinkley, 1997):

$$CI^j_{basic} = \left(2\hat{\eta}^j_n - \hat{\eta}^{j*}_{n,[B(1-\alpha/2)]},\ 2\hat{\eta}^j_n - \hat{\eta}^{j*}_{n,[B\alpha/2]}\right). \tag{4}$$

In our synthetic network example, $\hat{f}(3) = 0.208$, and its 95% bootstrap confidence interval (3), based on the 50 bootstrap samples, is

```
> CIpercentile <- boot_ci(bootdd)
> CIpercentile$fk_ci[,"3"]
      2.5%     97.5%
 0.1286842 0.2631579
```

The outputs of the functions `lsmi_dd`, `boot_dd`, and `boot_ci` are estimates of the network degree distribution $f(k)$ and mean degree μ based on a single patch. They are recognized as objects of class `'snowboot'` and can be plotted with the S3 method `plot` for this class (Figure 4).

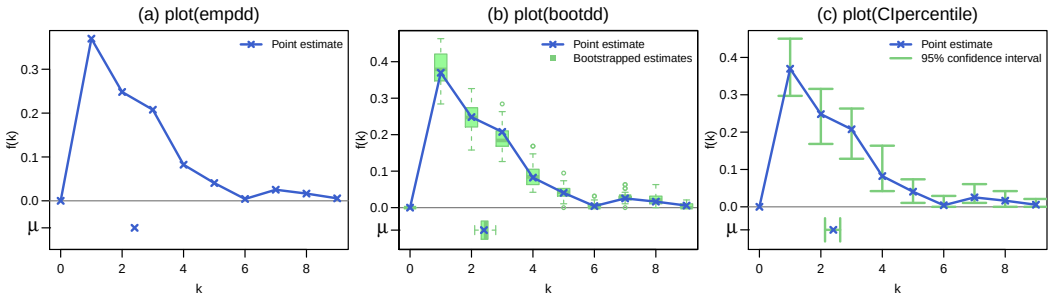

Figure 4: Results of plotting three different objects of class `'snowboot'`.

Cross-validation with LSMI

Growing snowball samples to higher waves is used in sampling surveys when obtaining new seeds is prohibitively expensive, for example, in hard-to-reach and "hidden" subpopulations of HIV high risk individuals. Even if no more new seeds can be obtained, varying the number of seeds in a patch offers an additional flexibility, so that multiple seed-wave combinations can be evaluated and an optimal seed-wave combination is selected for a given observed network.

Given bootstrap confidence intervals from patches for each jth seed-wave combination ($j = 1, \ldots, J$), we use Algorithm 2 of Gel et al. (2017) to decide which patch provides confidence intervals of the best coverage for our statistic η. We approximate the ground truth using proxy samples. A proxy sample is obtained by resampling (with or without replacement) vertices already used in the LSMIs, so no new information is needed from the network. From multiple such samples, proxy statistics $\hat{\eta}^{proxy}$ are estimated, and then for each j a proportion of proxy statistics within the interval CI^j is calculated. This proportion aims to approximate the true coverage probability of the bootstrap confidence intervals, so an interval $CI^{j_{opt}}$ with the coverage closer to the nominal level $1 - \alpha$ can be selected.

Cross-validation is performed automatically by an upper-level function `lsmi_cv` in **snowboot** package. This function obtains multiple patches for each of the specified seed-wave combinations and runs cross-validation to select the optimal bootstrap confidence interval based on proxy values for the mean (by default, 19 proxy samples are obtained, each comprising 30 vertices):

```
> cv <- lsmi_cv(net, n.seeds = c(10, 20, 30), n.wave = 5, B = 100)
> cv
  $`bci`
      2.5%    97.5%
  1.749167 2.952500

  $estimate
  [1] 2.387042

  $best_combination
  n.seed n.wave
      10      2

  $seeds
  [1]  218  323  851 1003 1039 1097 1410 1624 1723 1756
```

The output of `lsmi_cv` contains the optimal seed-wave combination (`best_combination`), the corresponding point estimate (`estimate`) and bootstrap confidence interval (`bci`), and, for information purposes, the IDs of the actual seeds that were used in the patch with the selected seed-wave combination (`seeds`). In this example, a fixed increment grid is used for the number of seeds and waves, however, a user may choose to do a stochastic search in the parameter space.

Why does patchwork bootstrap work and its areas of limitations

Bias vs. variance Many estimators of graph totals based solely on seeds are known to be unbiased (Frank, 1977). However, variance of such seed-based estimator might be high if the number of seeds is low. In turn, in many applications sampling more seeds might be prohibitively expensive, e.g., due to data privacy and cost restrictions (see overview by Illenberger and Flötteröd, 2012 and references therein). Adding information from non-seeds into the degree estimator increases bias but reduces

variance. Hence, the choice of optimal number of seeds (egos) and waves of non-seeds in LSMI implies a classical bias vs. variance trade-off, and we address it using the cross-validation procedure described above.

Network properties Furthermore, as in many non-network settings, in order to ensure consistency of the bootstrap estimator $\hat{\eta}_n^*$ of the statistic of interest η_n, based on a degree distribution $F_n = \{f_n(k), k \geq 0\}$ of a random graph G_n, we typically need to ensure that F_n is a satisfactory approximation of F as $n \to \infty$. That is, we need to ensure that the empirical degree distribution F_n cannot be drastically different from the population degree distribution F and shall approach the population degree distribution F with increasing n, which in turn is a necessary condition for $\eta_n \to \eta$ as $n \to \infty$. In addition, F needs to satisfy some invariance properties. Those conditions give rise to the assumptions for patchwork bootstrap on p. 2. Note that while nowadays there are no formal tests for assessing involution invariance, similarly as there exists no test to assess strong stationarity in time series, a class of involution invariant graphs includes, for example, such a large subclass as exchangeable graphs (Lovász, 2012; Orbanz and Roy, 2015). Asymptotic properties of patchwork bootstrap for $\hat{\eta}_n^*$ for involution invariant graphs are discussed in more details in Gel et al. (2017). In addition, Thompson et al. (2016) consider the two-phase conditional inference framework to derive asymptotic properties for patchwork bootstrap estimator of mean degree μ.

Sampling design Finally, we would like to emphasize that properties of bootstrap on networks largely depend not only on the underlying network topology but on the closely linked question on how sampling is performed. In this section we focus only on properties of a network degree distribution and argue that LSMI appears to be a suitable choice (see more discussion in Illenberger and Flötteröd, 2012; Gel et al., 2017). In turn, bootstrap and, generally, finite-sample inference for other network statistics will typically require adjustment of a sampling design (Kolaczyk, 2009).

Hence, the algorithms of LSMI sampling and estimation realized in **snowboot** package target such network statistics as network degree distribution (probabilities of observing vertices with a specific degree) and its functions (e.g., mean degree and network density). In turn, the algorithm of patchwork bootstrap aims to quantify the uncertainty of those estimates.

Switching to other network statistics, such as clustering coefficient or motifs, requires implementation of different sampling and bootstrap schemes, for example, via combination of patchwork bootstrap with the algorithm of Ali et al. (2016) (see Orbanz, 2017, for discussion on sampling and subsampling designs). Hence, this extension constitutes a natural direction for future methodological research and R code implementation.

Vertex bootstrap

Vertex bootstrap (Figure 5), or the Snijders–Borgatti procedure, is the pioneering approach to nonparametric bootstrap inference on graphs. Nevertheless, despite its high popularity in social sciences, vertex bootstrap remains largely unknown in statistics. Vertex bootstrap employs an induced graph sampling for quantification of standard errors in network density estimation and allows hypothesis testing on density of two networks. The algorithm assumes availability of the entire network data upfront as well as requires resampling of the entire data set and, hence, is limited to relatively small networks due to computational costs. To the best of our knowledge, there exists no analysis of asymptotic properties and theoretical guarantees for the Snijders–Borgatti procedure.

We implement vertex bootstrap in the R package **snowboot** as a function `vertboot`, which resembles calculations under the option "Network → Compare densities" of the UCINET software (Borgatti et al., 2002).

The `vertboot` function not only generates similar results compared with UCINET, but also returns results with higher precision and faster run times. In **snowboot**, the algorithm is written in C++.

Another important improvement is that `vertboot` allows users to compare statistics of interest for multiple networks, whereas the number of different networks, T (Figure 5), in UCINET is limited to 1 or 2. Finally, the output of `vertboot` (Figure 5) is a bootstrapped network, which implies that users can analyze various network statistics besides the network density.

We demonstrate the `vertboot` function using prison network data. The data were collected from 67 prison inmates; each inmate could choose as few or as many "friends" as he desired. A direct factor analysis of these sociometric data was performed by MacRae (1960).

- Get the observed adjacency matrix for the prison network:

```
> a <- scan("http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/prison.dat",
+           skip = 4)
> A <- matrix(a, sqrt(length(a)), byrow = TRUE)
```

Input : T observed adjacency matrices A_t $(t = 1, \ldots, T)$.
Output: T bootstrapped adjacency matrices A_t^*.

for t *in* $1, \ldots, T$ **do**
  $list = \{1, \ldots, nrow(A_t)\}$
  $list^* = sample\ with\ replacement\ elements\ of\ list$
  for i *in* $1, \ldots, length(list^*)$ **do**
    $ii = list^*[i]$
    for j *in* $1, \ldots, length(list^*)$ **do**
      $jj = list^*[j]$
      if $ii \neq jj$ **then**
        $A_t^*[i, j] = A_t[ii, jj]$
      end
      else
        $ii = random\ sample\ from\ list$
        $jj = random\ sample\ from\ list$
        while $ii = jj$ **do**
          $jj = random\ sample\ from\ list$
        end
        $A_t^*[i, j] = A_t[ii, jj]$
        $A_t^*[j, i] = A_t[ii, jj]$
      end
    end
  end
end

Figure 5: Vertex bootstrap.

- Apply vertex bootstrap (Figure 5) B times to generate B bootstrapped adjacency matrices (networks):

```
> set.seed(1)
> B <- 500
> Astar <- vertboot(A, B)
```

- Get the densities of B bootstrapped networks:

```
> library(igraph)
> densities <- sapply(1:B, function(x)
+    graph.density(graph_from_adjacency_matrix(Astar[[x]])))
```

- Estimate density of the observed network and bootstrap standard error of the estimates:

```
> density_obs <- graph.density(graph_from_adjacency_matrix(A))
> densities_se <- sd(densities)
```

- Obtain a 95% bootstrap confidence interval for the population density:

```
> quantile(densities, c(0.025, 0.975))
       2.5%      97.5%
0.03301673 0.05178652
```

We can use the outputs of the vertex bootstrap (Figure 5) to build bootstrap confidence intervals for other statistics:

- Mean degree:

```
> mu_star <- sapply(1:B, function(x)
+    mean(degree(graph_from_adjacency_matrix(Astar[[x]]), mode = "in")))
> quantile(mu_star, c(0.025, 0.975))
     2.5%    97.5%
2.179104 3.417910
```

- Transitivity:

```
> net_transitivity <- sapply(1:B, function(x)
+    transitivity(graph_from_adjacency_matrix(Astar[[x]]), type = "undirected"))
> quantile(net_transitivity, c(0.025, 0.975))
       2.5%      97.5%
0.08890533 0.21852252
```

Simulation studies

Light- and heavy-tailed degree distributions

In this section, we demonstrate the performance of the patchwork bootstrap algorithm for inference on mean degree of simulated graphs. We consider random graphs of orders 1000, 5000, and 10000 that are constructed from two polylogarithmic distributions of the same mean (population mean degree $\mu(G) = 2.67$ for all simulated networks), but with different tail behavior. Particularly, polylogarithmic distribution with parameters $\delta = 0.001$, $\lambda = 2.13$ has a light tail, whereas polylogarithmic distribution with parameters $\delta = 0.987$, $\lambda = 5$ has a heavy tail (Figure 6).

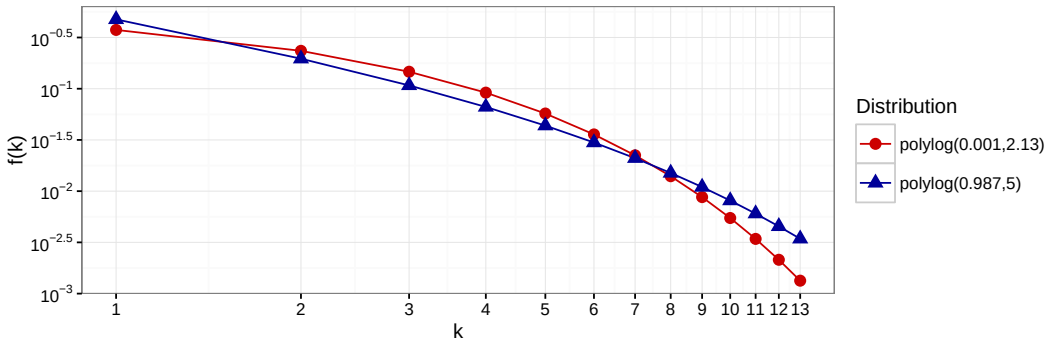

Figure 6: Polylogarithmic distributions with the same mean ($\mu = 2.67$), but different thickness of tails. The graphs are log-log graphs, where both axes use a logarithmic scale.

The results of our simulation study (Table 1) show a good performance of the patchwork bootstrap approach with automatic selection of patch sizes (i.e., seed-wave combinations) using the cross-validation procedure. Thus, the observed coverage probabilities of the confidence intervals are close to the nominal level 95% for both light- and heavy-tailed networks. At the same time, confidence intervals in the considered light-tailed networks are approximately 25% narrower. Time complexity of the patchwork procedure on any random graph is $\mathcal{O}([|d_s| \cdot \hat{\mu}_s]^{|d_{ns}|})$.

		Network order n		
Distribution	Mean degree $\mu(G)$	1000	5000	10000
polylog($\delta = 0.001$, $\lambda = 2.13$)	2.67	97.7% (0.794)	96.4% (0.812)	97.7% (0.805)
polylog($\delta = 0.987$, $\lambda = 5$)	2.67	98.6% (1.051)	98.7% (1.078)	97.7% (1.069)

Table 1: Observed coverage probabilities of 95% patchwork bootstrap confidence intervals for the mean degree (average interval width is in parentheses). These intervals come from the optimal seed-wave combination (one for each random graph) defined via a cross-validation over a grid of 20 combinations: waves from 1 to 5; and number of seeds 20, 30, 40, and 50. In cross-validation, proxy mean degrees are estimated 13 times from 100 vertices sampled without replacement from the patch data. The number of bootstrap samples, B, is 500. The experiment uses 1000 Monte Carlo simulations, carried out as parallel processes on a distributed computing cluster.

Vertex removal

In this section, we randomly remove vertices (and edges emanating from those vertices) from the simulated networks prior to applying the patchwork and vertex bootstrap methods for the inference on mean degree. This allows us to assess the methods' robustness and capabilities of providing reliable inference when part of the network information is missing. Our goal is to estimate when performance of the methods decreases significantly.

As Table 2 shows, when only 1% of the vertices are removed, both methods deliver confidence intervals with coverage close to the declared 95% level. When 2% of network vertices are removed, coverage probabilities of vertex bootstrap intervals decline to 81.8% and 89.8% for light- and heavy-tailed networks, respectively. The performance of vertex bootstrap further rapidly declines as more vertices are removed, and the decline is more severe for the light-tailed polylogarithmic distribution. Remarkably, patchwork bootstrap performs consistently well on both distributions (observed coverage probabilities are close to the nominal level), even when 5% of the vertices are removed (Table 2). Using

the example of the light-tailed polylogarithmic distribution, we show that coverage of vertex bootstrap declines to zero when 10% or 15% of vertices are removed, while respective coverage of the patchwork bootstrap intervals is 93.9% and 85.1%, respectively (Figure 7).

Distribution	Mean degree $\mu(G)$	Method	Number of vertices removed 1%	2%	3%	5%
polylog($\delta = 0.001$, $\lambda = 2.13$)	2.67	Patchwork	96.0% (0.807) [0.187]	97.8% (0.801) [0.194]	97.6% (0.806) [0.191]	97.5% (0.795) [0.196]
		Vertex bootstrap	96.3% (0.170) [0.090]	81.8% (0.170) [0.091]	54.7% (0.169) [0.090]	5.8% (0.169) [0.090]
polylog($\delta = 0.987$, $\lambda = 5$)	2.67	Patchwork	99.4% (1.072) [0.230]	99.0% (1.062) [0.239]	98.9% (1.062) [0.249]	99.4% (1.032) [0.237]
		Vertex bootstrap	96.3% (0.212) [0.113]	89.8% (0.211) [0.113]	74.0% (0.211) [0.113]	25.7% (0.209) [0.112]

Table 2: Observed coverage probabilities of 95% nonparametric bootstrap confidence intervals. Average interval width is in parentheses, standard errors are in square brackets. Network order $n = 5000$; number of bootstrap samples $B = 500$; 1000 Monte Carlo simulations. For the patchwork method, $J = 20$ seed-wave combinations: number of seeds 20, 30, 40, and 50; number of neighbors from 1 to 5.

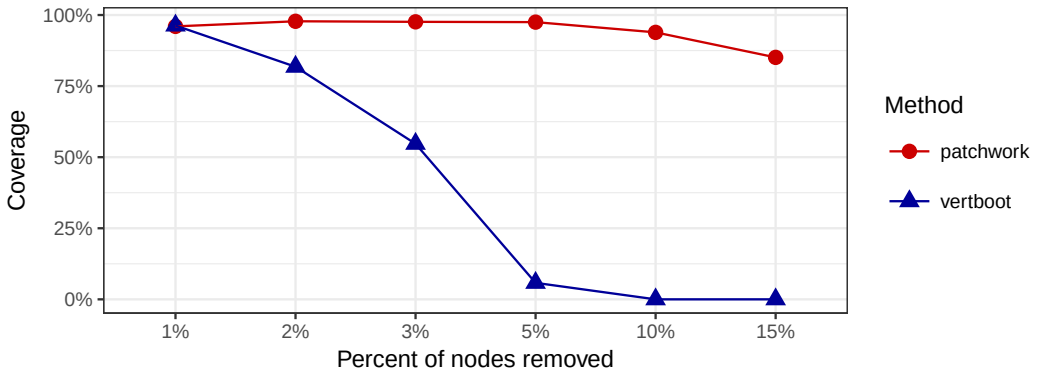

Figure 7: Observed coverage probabilities for 95% confidence intervals for network mean degree, delivered by the two bootstrap methods when different numbers of vertices are removed from the network. Network order $n = 5000$; degree distribution polylog($\delta = 0.001$, $\lambda = 2.13$); $B = 500$; 1000 Monte Carlo simulations.

Hence, the patchwork bootstrap approach is a competitive alternative when analyzing large complex networks, both in terms of computational speed and reliability of inference when a part of the network information is missing. Moreover, the patchwork method is both computationally efficient and information-greedy, i.e., only a small proportion of the network data is required, the procedure subsets the sampled seeds to consider patches of different sizes and re-uses information from the patches in the cross-validation procedure. In contrast, the vertex bootstrap employs information of the whole target network so that its time complexity is $\mathcal{O}(n^2)$. At the same time, for small networks with all the network information being available upfront, the vertex bootstrap is the preferred method as it provides noticeably sharper confidence intervals under the same level of calibration.

Case studies

In this section, we illustrate utility of the **snowboot** package for analysis of airline networks, power grids, and the David Copperfield network.

The David Copperfield network

We start from a smaller network, namely, the David Copperfield network collected by Newman (2006). It examines the lexicon of Charles Dickens's classic 19th century novel. The network vertices are common nouns and adjectives; undirected edges connect adjacent words (Figure 8).

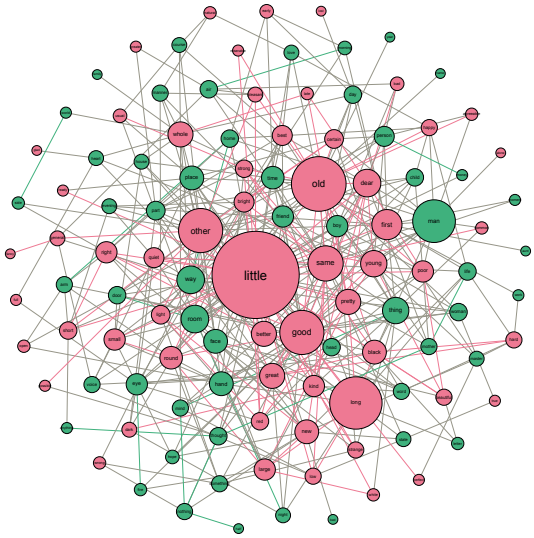

Figure 8: Lexical network: graph of nouns and adjectives found in the novel David Copperfield.

The number of vertices and edges in David Copperfield network are 112 and 425, respectively. This is a relatively small network, so the vertex bootstrap algorithm is suitable for analysis. Some basic network statistics of the David Copperfield network are presented in Table 3.

Order	Density	Number of edges	Mean degree	Clustering coefficient
112	0.068	425	7.589	0.157

Table 3: Parameters of the David Copperfield network.

Perform the vertex bootstrap in the following steps:

- Load the network data and obtain an adjacency matrix:

```
> library(igraph)
> graph_david <- read.graph(
+    "http://networkdata.ics.uci.edu/data/adjnoun/adjnoun.gml", format = "gml")
> A <- as.matrix(as_adjacency_matrix(graph_david))
```

- Use vertex bootstrap (Figure 5) to obtain bootstrapped adjacency matrices:

```
> library(snowboot)
> B <- 500
> set.seed(1)
> Astar <- vertboot(A, B)
```

- Use these bootstrapped networks to calculate 95% bootstrap confidence intervals for the density and bootstrap standard error:

```
> boot_density <- sapply(1:B, function(x)
+    graph.density(graph_from_adjacency_matrix(Astar[[x]])))
> CIvertboot <- quantile(boot_density, c(0.025, 0.975))
> CIvertboot
       2.5%      97.5%
 0.05473174 0.08579271
> bootstrap_standard_error <- sd(boot_density)
> bootstrap_standard_error
 [1] 0.007681788
```

That is, we find that the resulting 95% confidence interval from the vertex bootstrap is (0.055, 0.086); the interval contains the observed density of 0.068. Now let us apply the patchwork bootstrap to the David Copperfield network.

- Use the patchwork bootstrap to calculate 95% bootstrap confidence intervals for the density:

```
> set.seed(5)
> igraph_david <- igraph_to_network(graph_david)
> CIpatchwork <- lsmi_cv(igraph_david, n.seeds = c(3:5), n.wave = 1, B = B)
> CIpatchwork$bci/(igraph_david$n - 1)
       2.5%      97.5%
0.04305405 0.08957658
```

We find that the 95% confidence interval from the patchwork bootstrap also contains the observed density of 0.068. However, the patchwork bootstrap CI is wider than the vertex bootstrap CI (i.e., 0.047 for patchwork vs. 0.031 for vertex procedure). At the same time, the patchwork bootstrap used only about 31.4% of the available vertices.

Larger networks

Flight networks of the airline alliances

With constantly increasing costs of operation and rigid legal restrictions on ownership of national flag carriers, no single airline can provide a comprehensive global network, which is vital to its success. An urgent need for expansion propels airlines to effectively cooperate on multiple levels: from interlining (combining flights from different airlines in one travel itinerary) to joint frequent flyer programs, to sharing revenue, costs, and benefits (Pearce and Doernhoefer, 2011).

Nowadays, three major passenger airline alliances (Star Alliance, Oneworld, and SkyTeam) share more than 70% of the world market. There exists a constant, fierce competition in customer acquisition and retention among these three alliances, and one of its key factors is claimed to be a route map (number of served destinations and better connections). Traditional indicators, such as total passenger enplanements or number of aircraft movements, fail to capture the competitiveness of airline networks (Burghouwt and Redondi, 2013). However, for a fixed network structure, e.g., with hubs and spokes, dense air flight networks (with a high mean degree) are more convenient, since they further minimize the required number of connections. The International Air Transport Association (IATA) study of European Union countries showed that a 10% rise of the number of destinations served and/or the frequency of service can increase long-run gross domestic product by 1.1% (Smyth and Pearce, 2006).

We then evaluate densities of the three flight networks (Star Alliance, SkyTeam, and Oneworld) to assess which alliance offers the most convenient travel network to its passengers. Since flight connections are easier if a transfer occurs within one airport, we focus on airport networks. This approach treats all airports as separate vertices even some of them have the same service area (for example, Toronto Pearson International Airport and Billy Bishop Toronto City Airport).

To construct the networks, we obtained the crowdsourced air flights data from OpenFlights.org on October 23, 2013. At the time of analysis, Star Alliance, Oneworld, and SkyTeam included 28, 13, and 19 members, respectively. We used the airline codes to select all flights for each airline alliance, but removed self-loops and the cases with missing airport identifications (up to 0.4% of the records). To eliminate the repeated entries, including the codeshare flights within each alliance, we kept only unique links between airports. About 11.0% of all remaining vertices had different in-degree and out-degree, but we neglected the directions for simplicity. Thus, for each airline alliance, we obtained a network where the vertices stand for airports, and each unweighted undirected edge represents existing flight connection at least in one direction (Figure 9).

Table 4 reports the networks orders n and network densities $\hat{d}(G_n)$ estimated on all observed data along with the results of the patchwork bootstrap. The network order (i.e., number of served airports) is close to 1,000 for all three alliances. (Given a relatively high order of the airline networks, we do not consider the vertex bootstrap in this case study.) The optimal seed-wave combinations suggested by the cross-validation procedure differ among the networks: 20 seeds and 1 wave for SkyTeam, 30 seeds and 1 wave for Star Alliance, and 40 seeds and 1 wave for Oneworld. The width of the confidence interval for Star Alliance and Oneworld is only 0.00296 and 0.00321, respectively, compared with 0.00497 for SkyTeam. Since all three confidence intervals overlap with each other, we conclude that currently there exists no significant difference in flight connections offered by the three major airline alliances. Thus, the loyalty of frequent fliers and acquisition of new customers are likely to be attributed to other factors, such as customer service, loyalty program benefits, ticket prices, and availability of flights in particular regions (e.g., Figure 9 shows that Oneworld network provides almost no service in African airports).

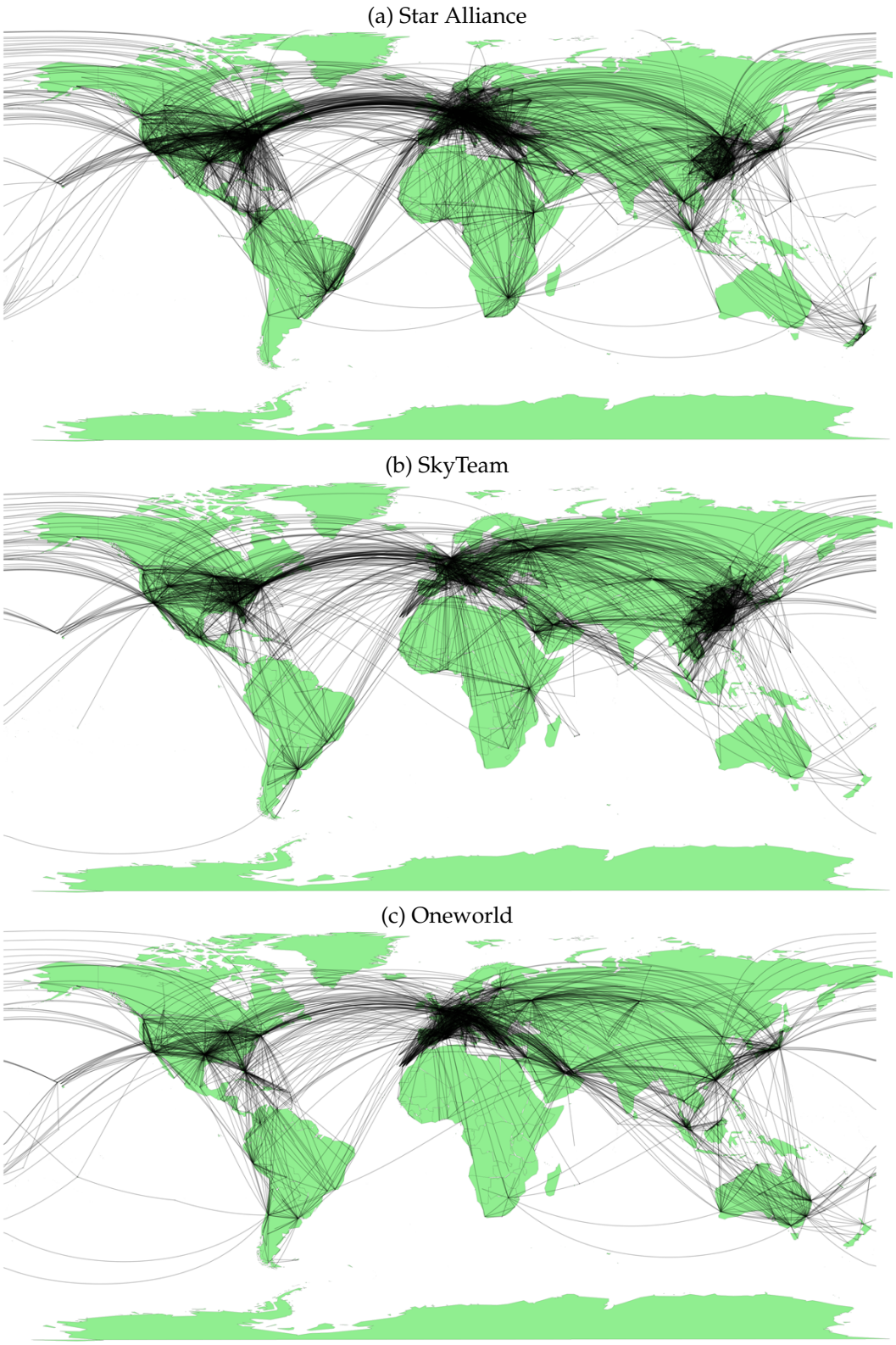

Figure 9: Airport networks of the airline alliances.

United States and Germany power grids

The power grid serves as the backbone of critical infrastructure sector and is essential for today's society as an enabling infrastructure. A combination of three substations, i.e., generator, transmission, and distribution, connected by high voltage transmission lines, provide the United States and Germany with electrical power people so heavily rely on. As with many other large scale infrastructures, the

			Optimal combination		95% confidence bounds for the density $d(G)$	
Network	n	$\hat{d}(G_n)$	Seeds	Waves	Lower	Upper
Star Alliance	1289	0.00621	30	1	0.00492	0.00788
SkyTeam	1040	0.00736	20	1	0.00561	0.01058
Oneworld	914	0.00655	40	1	0.00533	0.00854

Table 4: The 95% bootstrap confidence intervals for the density of airline alliance networks, replicating connections of the airports (vertices) with the flights of member airlines (edges). Considered 12 seed-wave combinations: waves from 1 to 3, seeds 20, 30, 40, and 50. Number of bootstrap resamples is 500 per each combination. Cross-validation is based on a random selection of 100 vertices 10 times.

power grid serves users who may notice its presence and realize its importance only when the system fails in some way. One of the main issues with the system is that failures or disruptive events like hurricanes, earthquakes, and attacks, can cause cascading failures in a power grid. As the power grids increase in size and complexity, it is of a paramount importance to study their vulnerability.

To better understand the effects of power system failures, the power grids can be analyzed from the perspective of random networks. In this case study, we consider two power grids that are represented as undirected networks, that is, the power grids of the western states of the United States (Watts and Strogatz, 1998) and Germany (Matke et al., 2016). Each edge represents a power supply line and vertex is a generator, a transformer, or a substation (Figure 10).

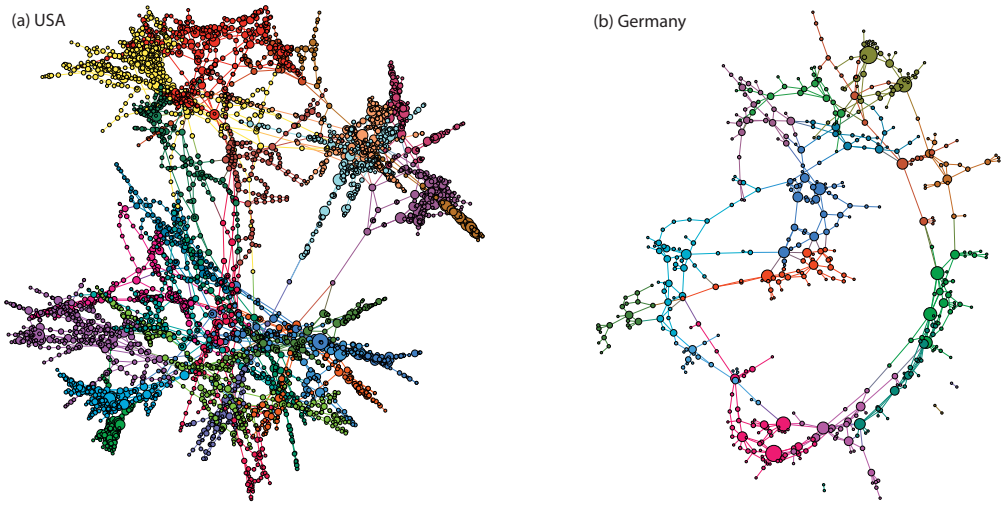

Figure 10: Power grids of the western states of the United States and Germany. The colors show network modules; connections between the vertices within modules are denser than between vertices from different modules.

Centrality statistics are one of the most widely explored attributes of a power grid network. Some studies focus on the relationship between various centrality statistics and resilience of the power grid networks (Pagani and Aiello, 2013). Another potential indicator of power system robustness and resilience is network density (Cuadra et al., 2015). In addition, Sóle et al. (2008) and Rosas-Casals and Corominas-Murtra (2009) propose to use a characteristic parameter γ, based on fitting an exponential distribution to an empirical cumulative distribution of each grid as a classifier of grid fragility. That is, the network is robust if $\gamma < 1.5$ and is fragile otherwise. In this study, we would like to examine the difference in fragility properties of the power grids in Germany and the western states of the United States, in terms of the γ parameter and the proportion of distribution stations in high-voltage networks. Given a relatively high order of the networks, the vertex bootstrap is not feasible, and we apply the patchwork bootstrap to compare the two power grids (see Table 5). We find that both power grids deliver the characteristic parameter $\hat{\gamma}$ of higher than 1.5, that is, 2.09 and 2.62 for the US and Germany, respectively, and hence both grids shall be classified as fragile. However, the respective 90% patchwork bootstrap confidence intervals do not overlap, and we can conclude that the US power grid in the western states tends to be less fragile than the German power grid. Remarkably, the 90% confidence intervals for densities of the two networks also do not overlap, hence, indicating that there likely exists a significant difference in connectivity of the two power grid networks.

While it is pre-mature to conclude that the Western US power grid system is more robust than the German power system due to its higher sparsity, especially given the lack of a uniformly accepted notion of power system fragility and robustness (Pagani and Aiello, 2013; Cuadra et al., 2015; Dey et al., 2017; Islambekov et al., 2018), it is reasonable to conclude that the two systems exhibit significant differences in their structure. In turn, the bootstrap methodology provides a route how power grids and their network properties can be systematically evaluated and compared in a framework of statistical hypothesis testing.

				90% confidence bounds			
				γ		$d(G)$	
Power grid network	n	$\hat{\gamma}$	$\hat{d}(G_n)$	Lower	Upper	Lower	Upper
United States	4941	2.08943	0.00054	1.75241	2.12021	0.00048	0.00056
Germany	523	2.62055	0.00642	2.20408	2.78395	0.00586	0.00692

Table 5: The 90% bootstrap confidence intervals for the fragility parameter γ and density $d(G)$ of two power grid networks. Considered 20 seed-wave combinations: waves from 1 to 5, seeds 20, 30, 40, and 50. Number of bootstrap resamples is 500 per each combination. Cross-validation is based on a random selection of 100 vertices 13 times.

Conclusion

In this paper we discuss utility and implementation of the two bootstrap methods for nonparametric inference on complex networks, that is, patchwork bootstrap of Thompson et al. (2016) and Gel et al. (2017) and vertex bootstrap of Snijders and Borgatti (1999). We primarily focus on developing inference on network degree distribution and its functions, e.g., mean and density. Furthermore, we perceive the observed network data as a single realization of some "true" unobserved network, and our target is to draw statistical inference in a model-free data-driven way, given only this single network realization. While there is an ever increasing interest in nonparametric inference on complex networks and despite the fact that the vertex bootstrap has been implemented in UCINET software for social network analysis for more than a decade, to our knowledge, there exists no single implementation of bootstrap methods on graphs in R. Our new package **snowboot** fills this gap and offers a flexible data-driven alternative for parametric analysis of complex networks. Furthermore, **snowboot** is fully compatible with **igraph**, and provides a number of options, such as Labeled Snowball Sampling with Multiple Inclusions and cross-validation on graphs – the functionality of standalone interest for graph mining and network analysis.

Acknowledgements

The authors thank Steve P. Borgatti for the help with the vertex bootstrap and Marti Rosas-Casals for the help with interpretation of γ fragility parameter in the power grid analysis. This work was partially supported by grants from the National Science Foundation (NSF) of the United States, IIS 1633331/1633355, DMS 1736368 and ECCS 1824710.

Bibliography

N. K. Ahmed, N. Duffield, J. Neville, and R. Kompella. Graph sample and hold: a framework for big-graph analytics. In *Proc. of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 1446–1455, 2014. URL http://dx.doi.org/10.1145/2623330.2623757. [p2]

D. Aldous and R. Lyons. Processes on unimodular random networks. *Electronic Journal of Probability*, 12(54):1454–1508, 2007. URL https://doi.org/10.1214/EJP.v12-463. [p2, 3]

W. Ali, A. E. Wegner, R. E. Gaunt, C. M. Deane, and G. Reinert. Comparison of large networks with sub-sampling strategies. *Scientific Reports*, 6:28955, 2016. URL https://doi.org/10.1038/srep28955. [p3, 8]

S. Bhattacharyya and P. J. Bickel. Subsampling bootstrap of count features of networks. *The Annals of Statistics*, 43(6):2384–2411, 2015. URL https://doi.org/10.1214/15-AOS1338. [p3]

S. P. Borgatti, M. G. Everett, and L. C. Freeman. *Ucinet 6 for Windows: Software for Social Network Analysis*. Analytic Technologies, Harvard, MA, 2002. URL https://sites.google.com/site/ucinetsoftware/home. [p1, 8]

G. N. Boshnakov. *Rdpack: Update and Manipulate Rd Documentation Objects*, 2018. URL https://CRAN.R-project.org/package=Rdpack. R package version 0.10-1. [p1]

T. Britton, M. Deijfen, and A. Martin-Löf. Generating simple random graphs with prescribed degree distribution. *J. of Statistical Physics*, 124(6):1377–1397, 2006. URL https://doi.org/10.1007/s10955-006-9168-x. [p2]

I. Brugere, B. Gallagher, and T. Y. Berger-Wolf. Network structure inference, a survey: Motivations, methods, and applications. *arXiv preprint arXiv:1610.00782*, 2017. URL http://arxiv.org/abs/1610.00782. [p1]

G. Burghouwt and R. Redondi. Connectivity in air transport networks: An assessment of models and applications. *J. of Transport Economics and Policy*, 47(1):35–53, 2013. URL http://www.jstor.org/stable/24396351. [p13]

M. R. Chernick. *Bootstrap Methods: A Guide for Practitioners and Researchers*. John Wiley & Sons, Hoboken, NJ, 2 edition, 2008. URL http://dx.doi.org/10.1002/9780470192573. [p1]

E. Crane. Steady state clusters and the Ráth–Tóth forest fire model. *arXiv preprint arXiv:1809.03462*, 2018. URL https://arxiv.org/abs/1809.03462. [p2, 3]

G. Csárdi and others. *Igraph: Network Analysis and Visualization*, 2018. URL https://CRAN.R-project.org/package=igraph. R package version 1.2.2. [p1]

L. Cuadra, S. Salcedo-Sanz, J. Del Ser, S. Jiménez-Fernández, and Z. W. Geem. A critical review of robustness in power grids using complex networks concepts. *Energies*, 8(9):9211–9265, 2015. URL http://dx.doi.org/10.3390/en8099211. [p15, 16]

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, Cambridge, 1997. URL https://doi.org/10.1017/CBO9780511802843. [p6]

A. K. Dey, Y. R. Gel, and H. V. Poor. Motif-based analysis of power grid robustness under attacks. In *Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on*, pages 1015–1019. IEEE, 2017. URL https://doi.org/10.1109/GlobalSIP.2017.8309114. [p16]

D. Eddelbuettel, R. Francois, J. J. Allaire, K. Ushey, Q. Kou, N. Russell, D. Bates, and J. Chambers. *Rcpp: Seamless R and C++ Integration*, 2017. URL https://CRAN.R-project.org/package=Rcpp. R package version 0.12.19. [p1]

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. URL http://dx.doi.org/10.1214/aos/1176344552. [p1]

S. Epskamp, D. Borsboom, and E. I. Fried. Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, 50(1):195–212, 2018. URL https://doi.org/10.3758/s13428-017-0862-1. [p3]

O. Frank. Structure inference and stochastic graphs. Technical Report AD0687176, Research Institute of National Defence Stockholm, 1968. [p2]

O. Frank. Estimation of graph totals. *Scandinavian Journal of Statistics*, 4(2):81–89, 1977. [p7]

Y. R. Gel, V. Lyubchich, and L. L. Ramirez Ramirez. Bootstrap quantification of estimation uncertainties in network degree distributions (a short version appeared as "Fast patchwork bootstrap for quantifying estimation uncertainties in sparse random networks" in SIGKDD MLG2016). *Sci. Rep.*, 7:5807, 2017. URL http://dx.doi.org/10.1038/s41598-017-05885-x. [p1, 2, 3, 5, 7, 8, 16]

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010. URL http://dx.doi.org/10.1561/2200000005. [p1]

L. A. Goodman. Snowball sampling. *The Annals of Mathematical Statistics*, 32(1):148–170, 1961. URL http://www.jstor.org/stable/2237615. [p2]

M. Granovetter. Network sampling: Some first steps. *American Journal of Sociology*, 81(6):1287–1303, 1976. URL http://doi.org/10.1086/226224. [p2]

P. Hall. *The Bootstrap and Edgeworth Expansion*. Springer-Verlag, New York, 1992. URL http://doi.org/10.1007/978-1-4612-4384-7. [p1]

J. Illenberger and G. Flötteröd. Estimating network properties from snowball sampled data. *Social Networks*, 34(4):701–711, 2012. URL https://doi.org/10.1016/j.socnet.2012.09.001. [p7, 8]

U. Islambekov, A. K. Dey, Y. R. Gel, and H. V. Poor. Role of local geometry in robustness of power grid networks. In *Global Conference on Signal and Information Processing (GlobalSIP), 2018 IEEE*, 2018. [p16]

E. D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer-Verlag, New York, 2009. URL http://doi.org/10.1007/978-0-387-88146-1. [p2, 8]

E. D. Kolaczyk and G. Csárdi. *Statistical Analysis of Network Data with R*, volume 65 of *Use R!* Springer-Verlag, New York, 2014. URL http://doi.org/10.1007/978-1-4939-0983-4. [p1]

L. Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012. [p2, 8]

D. Lusseau, H. Whitehead, and S. Gero. Incorporating uncertainty into the study of animal social networks. *Animal Behavior*, 75(5):1809–1815, 2008. URL https://doi.org/10.1016/j.anbehav.2007.10.029. [p3]

D. MacRae. Direct factor analysis of sociometric data. *Sociometry*, 23(4):360–371, 1960. URL http://doi.org/10.2307/2785690. [p8]

C. Matke, W. Medjroubi, and D. Kleinhans. SciGRID – An Open Source Reference Model for the European Transmission Network (v0.2), 2016. URL http://www.scigrid.de. [p15]

M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006. URL http://doi.org/10.1103/PhysRevE.74.036104. [p12]

P. Orbanz. Subsampling large graphs and invariance in networks. *arXiv preprint arXiv:1710.04217*, 2017. URL https://arxiv.org/abs/1710.04217. [p3, 8]

P. Orbanz and D. M. Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(2):437–461, 2015. URL https://doi.org/10.1109/TPAMI.2014.2334607. [p2, 3, 8]

G. A. Pagani and M. Aiello. The Power Grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013. URL http://doi.org/10.1016/j.physa.2013.01.023. [p15, 16]

B. Pearce and G. Doernhoefer. *The Economic Benefits Generated by Alliances and Joint Ventures. IATA Economics Briefing*. IATA Economics, 2011. [p13]

L. L. Ramirez Ramirez, K. Nezafati, Y. Chen, V. Lyubchich, and Y. R. Gel. *Snowboot: Bootstrap Methods for Network Inference*, 2018. URL https://CRAN.R-project.org/package=snowboot. R package version 1.0.0. [p1]

M. Rosas-Casals and B. Corominas-Murtra. Assessing European power grid reliability by means of topological measures. *WIT Transactions on Ecology and the Environment*, 121:527–537, 2009. URL http://doi.org/10.2495/ESUS090471. [p15]

J. Scott and P. J. Carrington. *The SAGE Handbook of Social Network Analysis*. SAGE Publications, Thousand Oaks, CA, 2011. URL http://dx.doi.org/10.4135/9781446294413. [p2]

J. Shao and D. Tu. *The Jackknife and Bootstrap*. Springer-Verlag, New York, 1995. URL http://dx.doi.org/10.1007/978-1-4612-0795-5. [p1]

O. Simpson, C. Seshadhri, and A. McGregor. Catching the head, tail, and everything in between: a streaming algorithm for the degree distribution. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 979–984. IEEE, 2015. URL http://dx.doi.org/10.1109/ICDM.2015.47. [p2]

M. Smyth and B. Pearce. *Airline Network Benefits. IATA Economics Briefing #3*. IATA Economics, 2006. [p13]

T. A. B. Snijders and S. P. Borgatti. Non-parametric standard errors and tests for network statistics. *Connections*, 22(2):161–170, 1999. [p1, 3, 16]

R. V. Sóle, M. Rosas-Casals, B. Corominas-Murtra, and S. Valverde. Robustness of the European power grids under intentional attack. *Physical Review E*, 77(2):026102, 2008. URL http://doi.org/10.1103/PhysRevE.77.026102. [p15]

M. E. Thompson, L. L. Ramirez Ramirez, V. Lyubchich, and Y. R. Gel. Using the bootstrap for statistical inference on random graphs. *Canadian Journal of Statistics*, 44(1):3–24, 2016. URL http://doi.org/10.1002/cjs.11271. [p1, 2, 3, 4, 5, 8, 16]

R. van der Hofstad. *Random Graphs and Complex Networks*. Cambridge University Press, Cambridge, 2017. URL https://doi.org/10.1017/9781316779422. [p2]

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998. URL http://doi.org/10.1038/30918. [p15]

T. W. Yee. *VGAM: Vector Generalized Linear and Additive Models*, 2018. URL https://CRAN.R-project.org/package=VGAM. R package version 1.0-6. [p1]

Y. Zhang, E. D. Kolaczyk, and B. D. Spencer. Estimating network degree distributions under sampling: An inverse problem, with applications to monitoring social media networks. *Annals of Applied Statistics*, 9(1):166–199, 2015. URL http://doi.org/10.1214/14-AOAS800. [p2]

Yuzhou Chen
Department of Statistical Science
Southern Methodist University
P.O. Box 750332
Dallas, Texas, 75275
USA
E-mail: yuzhouc@smu.edu

Yulia R. Gel, Kusha Nezafati
Department of Mathematical Sciences
University of Texas at Dallas
800 West Campbell Road
Richardson, Texas, 75080
USA
E-mail: ygl@utdallas.edu, kusha.nezafati@utdallas.edu

Vyacheslav Lyubchich
Chesapeake Biological Laboratory
University of Maryland Center for Environmental Science
146 Williams Street / P.O. Box 38
Solomons, Maryland, 20688
USA
E-mail: lyubchich@umces.edu