Subsampling of Complex Networks

Kai Rathmann

Supervisor: Prof. Dr. Thorsten Strufe
Coordinator: Benjamin Schiller

TU Darmstadt
FB 20 | Informatik

January 24, 2012

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Darmstadt University of Technology
Department of Computer Science
Software Technology Group (STG)

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in dieser oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, January 24, 2012 Kai Rathmann

Abstract

This thesis concerns itself with the problem of subsampling of complex networks. The questions we attempt to answer are: Which properties of the original network are preserved? To which extend are they preserved? How does this depend on the algorithm used, including its general properties and specific parameters? How does this depend on the nature of the original network?

We group known subsampling algorithms together, attempt a classification, and sketch new ones where gaps in the classification appear. We choose a two-dimensional classification based on neighbor selection and dimensional strategy. We end up with 22 viable algorithmic variants.

The network metrics we use are network size, degree distribution, diameter, characteristic path length, and clustering coefficient. The networks we subsample are Erdős–Rényi, Barabási–Albert, complete network, and three real-world networks. We carry out the evaluation using and extending a framework for which we newly implement all evaluated subsampling algorithms.

We are able to successfully link certain properties of subsampled networks to categories of subsampling algorithms (as opposed to only individual ones). Such category results include: Network size is primarily determined by neighbor selection and secondarily by dimensional strategy. Diameter and characteristic path length are best preserved overall by three variants of the Depth-First Search algorithm. Clustering coefficient is well preserved by subsampling algorithms choosing a finite but nonzero number of neighbors in every step. Algorithms choosing no or all neighbors produce extreme results.

Contents

1 Introduction

This thesis concerns itself with the problem of subsampling of complex networks. The questions we attempt to answer are: Which properties of the original network are preserved? To which extend are they preserved? How does this depend on the algorithm used, including its general properties and specific parameters? How does this depend on the nature of the original network?

The problem is interesting because in order to analyze complex networks one needs representative data sets, and in case of very large networks, e.g., social networks, a complete dataset cannot be processed[1] or is unavailable in the first place.

To our knowledge, existing papers on the subject only deal with one or very few subsampling algorithms each. In contrast, this thesis groups these known subsampling algorithms together, attempts a classification, and sketches new ones where gaps in the classification appear.

We carry out the evaluation using and extending the Graph-Theoretic Network Analyzer (GTNA), a framework introduced in [SBS+] and developed in the department where this thesis is written.[2] We newly implement all evaluated subsampling algorithms as part of this thesis.

This thesis deals with *static* network patterns. We do not analyze *evolutionary* network patterns, which describe networks as they grow and change through time. We limit our analysis to static network patterns because here, we can do straightforward comparisons between the original network and the subsampled network. The end results of this comparison may be applied elsewhere, too.

Moreover, this thesis deals with scenarios where we have access to the *full* network and can easily pick nodes at random. A different problem comes from the area of web-crawling or P2P networks, where the question is how to select a random node from a network if we only see a *neighborhood* [SRD+06].[3]

This thesis is structured as follows. In chapter 2, we introduce our terminology. In chapter 3, we describe network metrics which we later apply to subsampled networks in order to compare the results of various subsampling algorithms. Chapter 4 presents those algorithms. Chapter 5 details our implementation of those algorithms. In chapter 6, we evaluate the performance of the algorithms on various complex networks. We conclude with chapter 7.

[1] For example, according to [DR03], in studies of Internet routing protocols, computer communication researchers would like to do detailed simulations of the Border Gateway Protocol (BGP) or flow level simulations, but the simulations on networks with more than a few thousand nodes may be prohibitively expensive.

[2] See also `http://www.p2p.tu-darmstadt.de/research/gtna/`

[3] Of course, having access to the full network does not prevent us from also considering subsampling strategies that deliberately limit themselves to choosing among the neighbors of the current node.

2 Background

In this chapter, we introduce the terminology we use to discuss complex networks.

2.1 Complex Networks

A *complex network* $G = (V, E)$ consists of a set of nodes V and a set of edges $E \subseteq (V \times V)$. The complex network is *undirected* if the pairs $V \times V$ are unordered, and *directed* if they are ordered. If no attribute is given, the complex network is understood to be unordered.

We forbid self-loops (edges that start and end in the same node), so strictly $E \subset (V \times V)$. V and E are finite.

For an ordered pair $e = (v_1, v_2)$ in a directed complex network, e is an *outgoing edge* for v_1 (it *starts in* v_1) and is an *incoming edge* for v_2 (it *ends in* v_2).

For an (ordered or unordered) pair $e = (v_1, v_2)$ in a (directed or undirected) complex network, e is *incident on* v_1 and v_2, and v_1 and v_2 are *neighbors of* and *adjacent to* each other.

2.2 Node Degrees

The *degree* k_v of a node v is the number of edges incident on it. For a directed complex network, we differentiate between *in-degree* (number of incoming edges) and *out-degree* (number of outgoing edges). The degree of a complex network is the maximum degree of its nodes.

A network whose nodes all have the same degree is called *regular*.

2.3 Paths

A *path* is a finite sequence of nodes $v_1, v_2, v_3, ..., v_k$ such that the network contains an edge (v_i, v_{i+1}) for every $i \in \{1...k-1\}$. The definition applies to both undirected networks (with undirected edges) and directed networks (with edges outgoing for v_i and incoming for v_{i+1}). The *length* of a path is the number of its nodes minus one.

The *distance* between two nodes is the length of a shortest path between them.

3 Network Metrics

In this chapter, we present various metrics of complex networks. We use these metrics in our evaluation to compare the results of various subsampling algorithms with each other and with the original network.

3.1 Network Size

Definition

The *network size* is the number of nodes.

Known Properties

This metric is only interesting for subsampling algorithms where the resulting network size can actually vary. This is the case for what we call *revisiting* subsampling algorithms.

3.2 Degree Distribution

Definition

The *degree distribution* is the distribution of node degrees over the whole network.

For a directed complex network, we define *in-degree distribution* and *out-degree distribution* as the distribution of node in-degrees and out-degrees, respectively.

Known Properties

Certain network types have known degree distributions. For example, a random network, in which each of n nodes is connected (or not) with independent probability p (or $1-p$), has degree distribution $P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$.

Real-world networks often exhibit a degree distribution where most nodes only have few edges, but a few nodes, so-called hubs, have high degree. The degree distribution of such networks, which are called *scale-free networks*, obeys a Power Law: $P(k) \sim k^{-\gamma}$, where γ is a constant.

3.3 Diameter and Characteristic Path Length

Definition

The *diameter* is defined as the maximum distance between any two nodes in the complex network. The *characteristic path length* is defined as the average such distance.

Known Properties

So-called *small-world networks* have a low diameter and characteristic path length, i.e. any node is only a few hops away from any other node. The *small-world phenomenon* states that the network of human relationships forms such a small-world network.

This idea was introduced in 1967 by Stanley Milgram's famous experiment, described in [Mil67]. Individuals in the US cities of Omaha, Nebraska and Wichita, Kansas were informed by letter about a person in Boston, Massachusetts and asked to forward the letter to a personal acquaintance who knew the subject with higher probability. The letters that reached their destination had been forwarded about six times on average.

The term *six degrees of separation*[1] denotes the idea that a human being can reach any other in the world by about six of these steps. This has been experimentally verified in several recent cases. For example, [LH07] showed in 2007 that the average path length in Microsoft Messenger was 6.6, based on the data of 240 million users.

3.4 Clustering Coefficient

Definition

The definition of the *clustering coefficient* depends on the concept of *triplets*.

A triplet is three nodes that are connected by either two (open triplet) or three (closed triplet) undirected ties. A triangle (three nodes, each adjacent to the other two) consists of three closed triplets, one centered on each of the nodes.

The clustering coefficient, then, is defined as follows.

$$C = \frac{\text{number of closed triplets}}{\text{number of connected triplets of vertices}}$$

The clustering coefficient measures how much the nodes in a network cluster together.

Known Properties

According to [HL71] and [WS98], the nodes of most real-world networks tend to cluster together relatively tightly, more so than the average probability of an edge randomly created between two nodes.

[1] This term was not used by Milgram himself.

4 Subsampling Algorithms

In this chapter, we present known algorithms, attempt a classification, and sketch new ones where gaps in the classification occur. We look at two classes of algorithms: Random walk algorithms and network traversal algorithms.

Starting at some node, a random walk will choose a neighbor of that node according to some criterion, visit that node, and then repeat the process until a budget of steps is exhausted. The same node can well be visited more than once by a random walk.[1]

Network traversal algorithms are often intended to address the problem of visiting *all* the nodes in a network in a particular manner. The next node to be visited need not be adjacent to the last visited node, so no node is visited more than once. The main application of this is search in a network. By terminating the algorithm after a number of steps, we can include network traversal algorithms for our purpose of subsampling.

Finally, there is an algorithm that does not fit into either category.

4.1 Random Walk Algorithms

Random walk algorithms have well-known properties, described in [GKBM10]. On the positive side, they normally require fewer resources per sample. On the negative side, they can suffer from large estimation errors in the presence of disconnected or loosely connected networks.

According to [LF06], random walks are among the best performing subsampling algorithms, with sample sizes down to about 15% of the original network.

4.1.1 Classic Random Walk

In the *Classic Random Walk* (RW), described in [Lov93], the next-hop node w is chosen uniformly at random among the neighbors of the current node v. That means the probability of moving from v to w is

$$P_{v,w}^{RW} = \begin{cases} \frac{1}{k_v} & \text{if } w \text{ is a neighbor of } v, \\ 0 & \text{otherwise.} \end{cases}$$

This random walk exhibits a bias toward high-degree nodes. The probability of being at the particular node v converges to the stationary distribution $\pi_v^{RW} = \frac{k_V}{2 \cdot |E|}$.

The bias can be potentially corrected by modifying the transition probabilities.

[1] If a random walk algorithm is modified with the restriction that no node is visited more than once, this is called a self-avoiding walk (SAW). We do not evaluate SAWs as a distinct group. Instead, two variants of the depth-first search algorithm introduced in 4.4 named $\text{DFS}_{\text{uniform}}$ and $\text{DFS}_{\text{weighted}}$ fill this role for the (classic) random walk, for example.

4.1.2 Metropolis-Hastings Random Walk

The *Metropolis-Hastings Random Walk* (MHRW), introduced in [MRR+53], attempts to correct the bias of the Classic Random Walk. It appropriately modifies the transition probabilities so that it converges to the desired uniform distribution.

The algorithm uses the following transition probability:

$$P_{v,w}^{MH} = \begin{cases} \frac{1}{k_v} \cdot \min(1, \frac{k_v}{k_w}) & \text{if } w \text{ is a neighbor of } v, \\ 1 - \sum_{y \neq v} P_{v,y}^{MH} & \text{if } w = v, \\ 0 & \text{otherwise.} \end{cases}$$

Other algorithms have been based on MHRW.[2]

4.1.3 Respondent-Driven Sampling

At each iteration, *Respondent-Driven Sampling* (RDS), described in [KMT10], randomly selects n (typically 3) neighbors of the current node and schedules them to visit later. RDS visits the nodes in the order they were scheduled. Thus, RDS is a modification of Snowball Sampling (see 4.2.5) that allows node revisiting. RDS introduces a known degree bias that can be corrected for. For $n = 1$, RDS becomes the classic random walk.

4.1.4 Frontier Sampling

Frontier Sampling (FS), introduced in [RT10], is a multidimensional dependent random walk. Let m be the number of (classic) random walks. They are initialized by picking a uniformly random node for each. Then, in each step, one random walk is picked to advance one step. The probability of a particular random walk to be picked is the degree of its current node divided by the sum of the degrees of the current nodes of all random walks.

4.2 Network Traversal Algorithms

In every iteration, a network traversal algorithm visits a node and discovers all its neighbors. How the algorithm proceeds depends on the particular sampling method.

4.2.1 Breadth-First Search

Breadth-First Search (BFS) proceeds by selecting the earliest explored but not yet visited node.

BFS has the property that even an incomplete sample collects a full view (all nodes and edges) of some particular region in the network.

According to [KMT10], BFS exhibits a bias toward high-degree nodes. The bias has not been analyzed, however.

[2] For example, the Metropolized Random Walk with Backtracking, introduced in [SR06], is a modified MHRW to deal with peer churn in peer-to-peer (P2P) networks. As we are not considering such networks in this thesis (see chapter 1), we disregard this algorithm.

4.2.2 Depth-First Search

Depth-First Search (DFS) proceeds by selecting the earliest explored but not yet visited node, but only among those added in the most recent iteration. If there are none, only then it considers nodes added one iteration before, etcetera.

4.2.3 Iterative Deepening Depth-First Search

Iterative Deepening Depth-First Search (IDDFS) is a variant of DFS. In the i-th iteration it will perform DFS up to a depth of i, always starting from the same origin. In every iteration the process thus revisits the nodes visited in previous iterations[3], but the cumulative order in which nodes are first visited is effectively BFS.

4.2.4 Forest Fire

Forest Fire (FF), described in [KMT10], is a randomized version of BFS where for every neighbor v of the current node, we flip a coin, with probability of success p, to decide if we explore v. FF reduces to BFS for $p = 1$.

It is possible that this process dies out before it covers all nodes. To continue, the process can be revived from a random node already in the sample.

4.2.5 Snowball Sampling

Snowball Sampling (SS), introduced in [Goo61], is a precursor of RDS. The term is loosely used for BFS-like traversal techniques. We define SS as follows: At every node v, not all k_v neighbors of v, but exactly n unvisited neighbors are chosen randomly.

4.3 Uniform Node Sampling

Uniform Node Sampling (UNS) means to uniformly at random select a set of nodes V, and a sample is then the network induced by V.

If a list of all nodes is not available and nodes have to be guessed, for example among a user ID space, then node sampling is undesirable if said ID space is sparsely populated [GKBM10]. We do not face this problem, as our networks will all be loaded into GTNA where a list of all nodes is available.

UNS does not retain power-law degree distribution; see [RT10] and [SWM05], for example.

4.4 Classification of Subsampling Algorithms by Concept

4.4.1 Initial Classification of Existing Algorithms

So far, we distinguished between random walks and network traversal algorithms. Talking about underlying concepts, these will henceforth be referred to as *revisiting* and *non-revisiting* instead. UNS is non-revisiting. IDDFS is revisiting.

[3] This is the only exception among our chosen network traversal algorithms where revisiting occurs. We take this into account in our later classification in 4.4.

Another noticeable distinction is how many *neighbors* of a node are selected in every step, denoted by the parameter n. Some algorithms select a single neighbor (e.g., RW and MHRW); some select several neighbors (SS and RDS); some select all neighbors (BFS). We define UNS as selecting no neighbors, and FF as selecting all neighbors. Each algorithm thus has exactly one of the properties $n = 0$, $n = 1$, $1 < n < \infty$ and $n = \infty$. We have arrived at a two-dimensional distinction.

We can further distinguish by how the *selection* of neighbors takes place (e.g., in storage order, uniformly random or weighted). Finally, to account for FS, we need to distinguish between one-dimensional and *multidimensional revisiting* algorithms. (We have also considered multidimensional *non*-revisiting algorithms, but those do not seem to add much.)

The resulting tabulation of concepts and algorithms is shown in Table 4.1.

Neighbors	Selection	non-revisiting	revisiting	multidimensional revisiting	
				independent	dependent
$n = 0$	uniform	UNS			
$n = 1$	ordered	DFS	IDDFS		
	uniform		RW		FS
	weighted		MHRW		
$1 < n < \infty$	ordered				
	uniform	SS	RDS		
	weighted				
$n = \infty$, $p = 1$	all	BFS			
$n = \infty$, $0 < p < 1$	p fixed	FF			
	p weighted				

Table 4.1.: Initial classification of existing algorithms

In order to fit all the algorithms into the table, entries in the same column may employ different ways to restart the process if it dies out. This may sometimes be an unorthodox way of viewing the algorithm in question:

- DFS needs to be thought of as dying out whenever it reaches a dead-end, and its restarting strategy is backtracking.
- UNS need to be thought of as dying out in every step ($n = 0$, so no neighbor is selected); in essence, its restarting strategy of picking a uniformly random node is all there is to the algorithm in this view.

4.4.2 Investigated Combinations of Concepts

Table 4.1 reveals many gaps where combinations of existing concepts into new algorithmic variants are conceivable. But not all of those variants are viable:

- For UFS, the revisiting and multidimensional variants do not make a meaningful difference to the algorithm.
- IDDFS, while a creative fit in the table, results in the same visited nodes as BFS and is therefore excluded in the final tabulation. Because multidimensional non-revisiting algorithms (such as a multidimensional BFS) are not included, the multidimensional variants of IDDFS are likewise dismissed.

- For BFS, node revisiting does not alter the outcome except for less nodes visited. For FF and small values of p, the outcome may change by occasionally "second-guessing" the coin-flip. Neither of these appear to be very interesting variants.

The remaining algorithmic variants are viable for evaluation. We introduce a naming scheme that makes them easy to identify, shown in Table 4.2. To talk about a set of related algorithms, we introduce a simplified naming scheme, shown in Table 4.3.

Neighbors	Selection	non-revisiting	revisiting	multidimensional revisiting	
				independent	dependent
$n = 0$	uniform	UNS	–	–	–
$n = 1$	ordered	$\text{DFS}_{\text{ordered}}$	–	–	–
	uniform	$\text{DFS}_{\text{uniform}}$	$\text{RW}_{\text{uniform}}$	$\text{RW}^{\text{multi}}_{\text{uniform}}$	$\text{FS}_{\text{uniform}}$
	weighted	$\text{DFS}_{\text{weighted}}$	$\text{RW}_{\text{weighted}}$	$\text{RW}^{\text{multi}}_{\text{weighted}}$	$\text{FS}_{\text{weighted}}$
$1 < n < \infty$	ordered	$\text{SS}_{\text{ordered}}$	–	–	–
	uniform	$\text{SS}_{\text{uniform}}$	$\text{RDS}_{\text{uniform}}$	$\text{RDS}^{\text{multi}}_{\text{uniform}}$	$\text{FS}^{1<n<\infty}_{\text{uniform}}$
	weighted	$\text{SS}_{\text{weighted}}$	$\text{RDS}_{\text{weighted}}$	$\text{RDS}^{\text{multi}}_{\text{weighted}}$	$\text{FS}^{1<n<\infty}_{\text{weighted}}$
$n = \infty$, $p = 1$	all	BFS	–	–	–
$n = \infty$, $0 < p < 1$	p fixed	FF_{fixed}	–	–	–
	p weighted	$\text{FF}_{\text{weighted}}$	–	–	–

Table 4.2.: Investigated combinations of concepts

Neighbors	non-revisiting	revisiting	multidimensional revisiting	
			independent	dependent
$n = 0$	UNS	–	–	–
$n = 1$	DFS	RW	RW-multi	FS
$1 < n < \infty$	SS	RDS	RDS-multi	FS+
$n = \infty$, $p = 1$	BFS	–	–	–
$n = \infty$, $0 < p < 1$	FF	–	–	–

Table 4.3.: Simplified naming scheme

Some of these combinations of concepts require further explanation:

- BFS is traditionally terminated using a *time to live* (TTL). Yet all other algorithms we included in our final tabulation are terminated when a budget of visited nodes is exhausted. Thus we use BFS with a budget system instead.

- $\text{FF}_{\text{weighted}}$ modifies the coin flip in the standard FF (FF_{fixed}) such that the probability of success is no longer p, but $p \cdot \min(1, \frac{k_v}{k_w})$, where k_v is the degree of the current node and k_w is the degree of the neighbor under consideration.

- Multidimensional independent algorithms are called in round-robin order. The budget is shared between all instances.[4]

[4] The number of dimensions will of course be chosen to be much smaller than the total budget. The difference between round-robin order and uniformly random selection is not interesting in that case.

5 Implementation

In this chapter, we introduce our implementation of the subsampling algorithms in GTNA, and explain our usage of metrics and networks with this framework.

5.1 Subsampling Algorithms

GTNA had not yet included any subsampling algorithms. We newly implemented the following algorithms:

- Uniform Node Sampling (UNS),
- Depth-First Search ($DFS_{ordered}$, $DFS_{uniform}$ and $DFS_{weighted}$),
- Snowball Sampling ($SS_{ordered}$, $SS_{uniform}$ and $SS_{weighted}$),
- Breadth-First Search (BFS),
- Forest Fire (FF_{fixed} and $FF_{weighted}$),
- Classic Random Walk ($RW_{uniform}$),
- Metropolis-Hastings Random Walk ($RW_{weighted}$),
- Respondent-Driven Sampling ($RDS_{uniform}$ and $RDS_{weighted}$),
- multidimensional independent Classic Random Walks ($RW^{multi}_{uniform}$),
- multidimensional independent Metropolis-Hastings Random Walks ($RW^{multi}_{weighted}$),
- multidimensional independent Respondent-Driven Samplings ($RDS^{multi}_{uniform}$ and $RDS^{multi}_{weighted}$), and
- Frontier Sampling ($FS_{uniform}$, $FS_{weighted}$, $FS^{1<n<\infty}_{uniform}$ and $FS^{1<n<\infty}_{weighted}$).

GTNA had already included support for *transformations*, which alter a complex network in some way. Apart from calling the super constructor with their own name and parameters, new transformations have to override two abstract methods:

- `applicable(Graph) : boolean`
  returns `true` if the transformation can be applied to the argument.
- `transform(Graph) : Graph`
  applies the transformation to the argument.

We introduced a new abstract class for subsampling algorithms as a subclass of `Transformation`. Apart from, again, calling the super constructor with their own name and parameters, new subsampling algorithms have to override one abstract method:

- `subsample(Graph) : Collection<Node>`
  applies the subsampling algorithm to the argument.

The abstract `Subsampling` class implements some more methods to normalize the result obtained by `subsample()` so that subclasses do not need to implement normalization. Normalization includes:

- Removing edges if the node they start or end in was not selected during the subsampling.
- Reassigning nodes indices. In GTNA, each node has a unique index, and the n nodes of a complex networks have indices between 0 and $n - 1$. Subsampling leaves gaps among the indices that are filled by reassignment.

5.2 Network Metrics

GTNA includes the metrics described in chapter 3 (network size, degree distribution, diameter, characteristic path length, clustering coefficient). The diameter and the characteristic path length are part of the Shortest Paths metric in GTNA.

5.3 Networks

We use network models and real-world networks in our evaluation.

5.3.1 Network Models

There are generators for various network models in GTNA, including those we use in our evaluation. The generated networks are written to the hard drive and reimported for the subsamplings using GTNA's `ReadableFolder` and `ReadableFile` classes.

5.3.2 Real-World Networks

The real-world networks we evaluate are imported using GTNA's `ReadableFile` class.

6 Evaluation

In this chapter, we evaluate the performance of the subsampling algorithms on various complex networks.

6.1 Setup

We use the following setup for our evaluation.

6.1.1 Number of Runs

We execute 100 runs on every configuration, that is, on every pair of network (model) and subsampling algorithm configuration. In case of network models, where we initially generate 100 networks that we reuse for each subsampling algorithm, this means that we run the algorithm once on each of the 100 networks. In case of the real-world networks, we run the algorithm 100 times on the same network.

For each metric, we calculate the average value across the 100 subsampled networks.

6.1.2 Budget

All subsampling algorithms are run with a budget of 10%, which is the higher of two values used in [RT10]. That paper also uses 1%, but the networks used there are orders of magnitude larger than ours. With our network sizes of down to less than 10,000 nodes, having subsampled networks of less than 100 nodes would not allow for granular enough measurements.

For revisiting algorithms, the actual percentage of distinct nodes sampled will be lower than the budget of 10%. This brings up the question of how to compare the values produced by two revisiting subsampling algorithms when the subsampled networks are of different size and the metric under evaluation is affected by the network size.

In such a case, we rerun the subsampling algorithm that has produced the larger network and terminate it early once it has collected as many nodes as the other algorithm.

6.1.3 Parameters

SS and RDS are run with parameter $n = 3$ which is the typical value quoted in [KMT10].

FF is run with parameter $p = 0.7$ which is the optimum value quoted in [LF06] for static network patterns.

Multidimensional algorithms are run with $m = 10$ instances which is the lowest of three values used in [RT10]. That paper also uses $m = 100$ and $m = 1000$, but again, the networks used there are orders of magnitude larger than ours. With our network sizes of down to less than 10,000 nodes, each instance of a multidimensional algorithm would have a miniscule budget using values for m that high.

6.1.4 Clarification on Edges

The framework GTNA models undirected networks as directed networks in which, for every edge $e = (v_1, v_2)$, there exists an edge $e' = (v_2, v_1)$. Hence, what would be a single (undirected) edge according to

our definition in chapter 2 is counted twice in GTNA's (total) degree distribution. We obtain the proper values by using the out-degree distribution instead.[1]

6.2 Hypotheses

For all metrics, our hypotheses on the results of BFS and UNS are based on the fact that the former collects a full view of some particular region in the network, whereas the latter is completely devoid of focus on any particular region in the network. Moreover, the $1 < n < \infty$ and FF variants will, in general, collect more densely than the $n = 1$ variants.

6.2.1 Network Size

We expect algorithms with $1 < n < \infty$ to yield lower network sizes than those with $n = 1$, especially in real-world networks, because selecting more neighbors will result in a more dense view of the network, with a higher likelihood in each step to select a neighbor that is already in the sample.

6.2.2 Degree Distribution

To begin with, we expect to reproduce the known result that $\mathrm{RW}_{\mathrm{uniform}}$ is more shifted toward high-degree nodes than $\mathrm{RW}_{\mathrm{weighted}}$ and $\mathrm{FS}_{\mathrm{uniform}}$.

We expect our concept combination $\mathrm{FS}_{\mathrm{weighted}}$ to be even less shifted toward high-degree nodes than $\mathrm{RW}_{\mathrm{weighted}}$ and $\mathrm{FS}_{\mathrm{weighted}}$.

We expect our concept combination $\mathrm{SS}_{\mathrm{weighted}}$ to be less shifted toward high-degree nodes than $\mathrm{SS}_{\mathrm{ordered}}$ and $\mathrm{SS}_{\mathrm{uniform}}$.

We expect BFS to retain the highest-degree nodes of all subsampling algorithms, and that BFS is followed in this capacity by FF, and that FF in turn is followed by algorithms with $1 < n < \infty$. We expect UNS to retain the least high-degree nodes.

6.2.3 Diameter and Characteristic Path Length

We expect BFS to produce the lowest diameters and characteristic path lengths, and UNS to produce the highest. In between, we expect the SS variants to produce lower values than the DFS variants.

6.2.4 Clustering Coefficient

We expect BFS to produce the highest clustering coefficients, and UNS to produce the lowest.

[1] The in-degree distribution would work as well.

6.3 Network Models

Network models are construction specifications with which one can generate networks with certain properties.

6.3.1 Erdős–Rényi

The *Erdős–Rényi* model, named for Paul Erdős and Alfréd Rényi, is a model for generating random networks that sets an edge between each pair of nodes with equal probability, independently of the other edges.

We generate 100 Erdős–Rényi networks with 10,000 nodes each and an average degree of 50. We end up with the following average values: A diameter of 4.0, a characteristic path length of 2.77, and a clustering coefficient of 0.0050. The average out-degree distribution is shown in Figure 6.1.

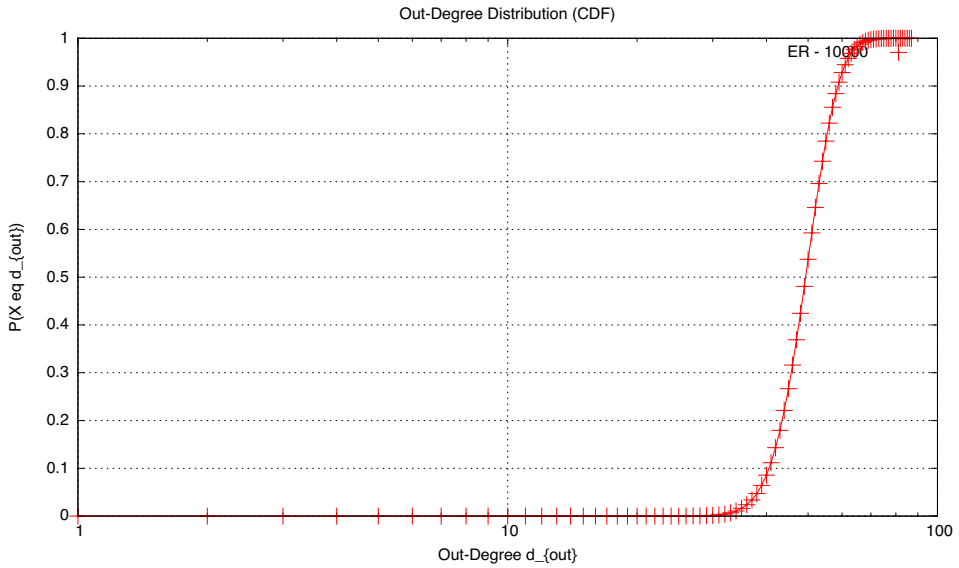

Figure 6.1.: Out-degree distribution (cumulative) for ER

6.3.2 Barabási–Albert

The *Barabási–Albert* model, named for Albert-László Barabási and Réka Albert, is a model for generating random scale-free networks.

We generate 100 Barabási–Albert networks with 10,000 nodes each and 25 edges per node. We end up with the following average values: A diameter of 2.3, a characteristic path length of 2.00, and a clustering coefficient of 0.4592. The average out-degree distribution is shown in Figure 6.2.

6.3.3 Complete Network

A *complete network* is a network where every node has an edge to every other node.

We generate one complete network with 1,000 nodes.

The only metric for which this is interesting is network size, because all subsampled networks are the same except for the size, which can be less than 10% of the original network in case of revisiting sub-

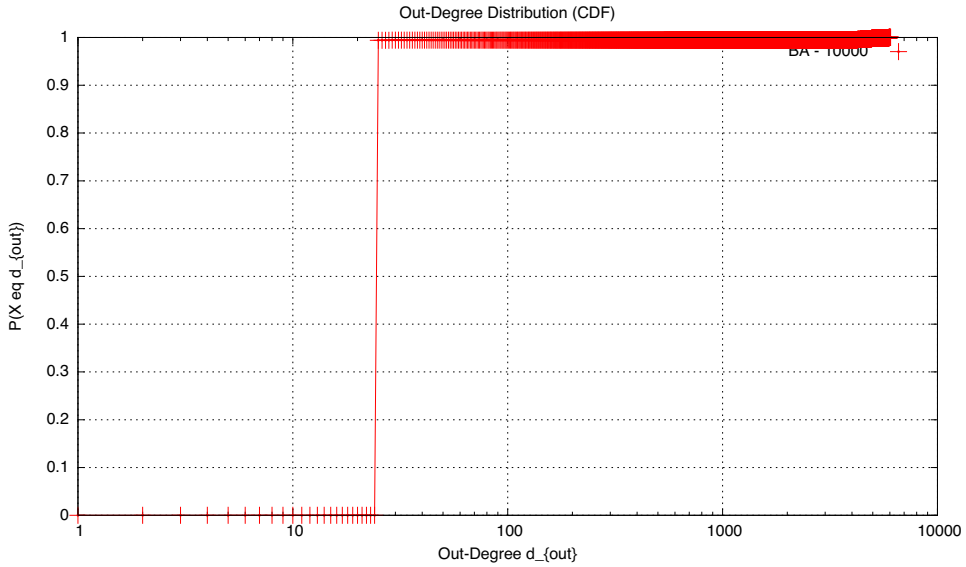

Figure 6.2.: Out-degree distribution (cumulative) for BA

sampling algorithms. But since every node has the same degree and has an edge to every other node in the graph, on average all revisiting subsampling algorithms perform almost exactly the same.[2]

6.4 Real-World Networks

In addition to network models, which have idealized properties, we include some real-world networks in our evaluation.

6.4.1 Studentenportal Ilmenau

Studentenportal Ilmenau (SPI) is a social network of students of Ilmenau University of Technology, Germany.

The network under consideration consists of 9,223 nodes and is undirected. It has a diameter of 12, a characteristic path length of 4.66, and a clustering coefficient of 0.2964. The out-degree distribution is shown in Figure 6.3.

6.4.2 Web of Trust

Web of Trust (WOT) is a community-based website reputation rating tool that uses a traffic-light style rating system to give Internet users additional information about a website before they visit it.

The network under consideration consists of 25,487 nodes and is available both in a directed and an undirected version.

[2] They do not perform *exactly* the same. RW, RW-multi and FS can theoretically return networks of 2 nodes by going back and forth between the starting node and one other node. RDS, RDS-multi and FS+ must select n distinct neighbors and can therefore only return networks of at least $n+1$ nodes.

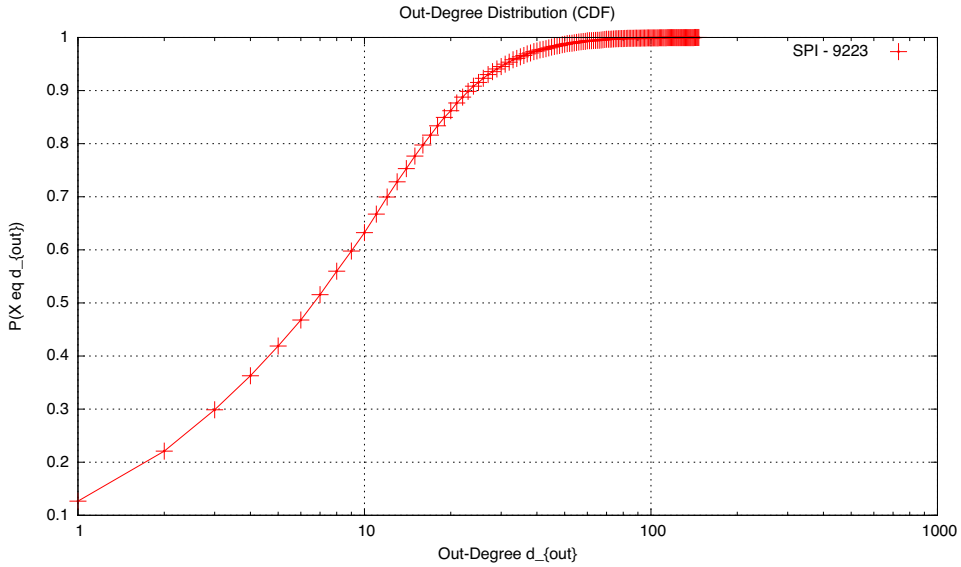

Figure 6.3.: Out-degree distribution (cumulative) for SPI

Undirected Version

The undirected version has a diameter of 15, a characteristic path length of 5.07, and a clustering coefficient of 0.4665. The out-degree distribution is shown in Figure 6.4.

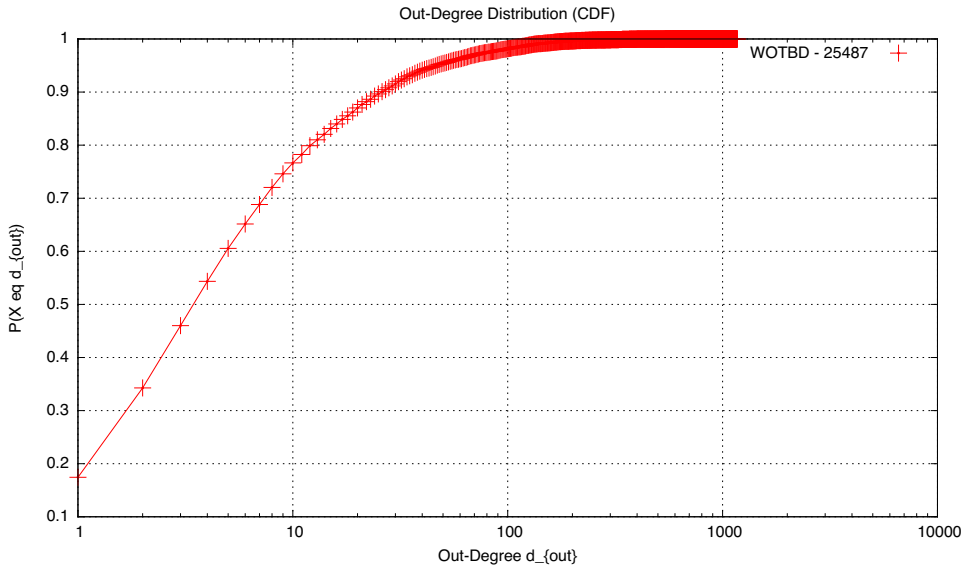

Figure 6.4.: Out-degree distribution (cumulative) for WOT, undirected

Directed Version

The directed version has a diameter of 25, a characteristic path length of 5.99, and a clustering coefficient of 0.3707. The in- and out-degree distributions are shown in Figure 6.5 and Figure 6.6.

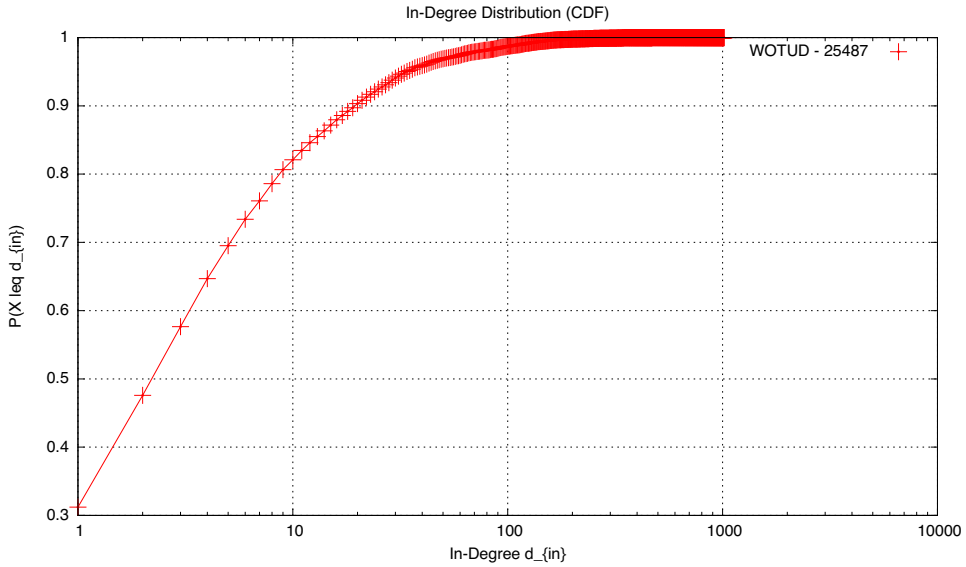

Figure 6.5.: In-degree distribution (cumulative) for WOT, directed

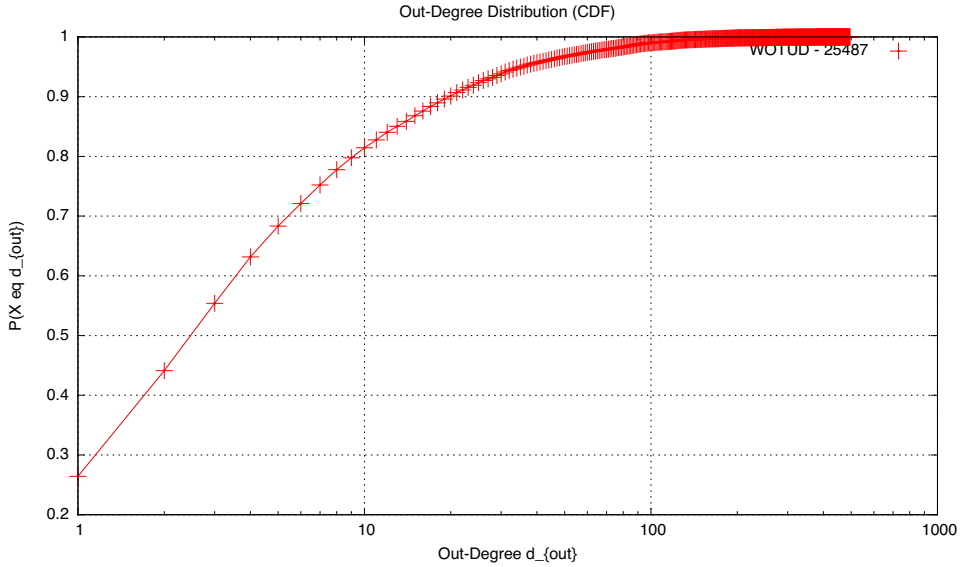

Figure 6.6.: Out-degree distribution (cumulative) for WOT, directed

6.5 Results

In the following discussion of our results, for the sake of a more readable presentation, we may write something like "the diameter produced by $\mathrm{DFS}_{\mathrm{ordered}}$" when we are talking about the *average* diameter *of the subsampled networks* produced by $\mathrm{DFS}_{\mathrm{ordered}}$.

6.5.1 Network Size

The resulting network size of revisiting (one- or multidimensional) subsampling algorithms is strongly affected by the parameter n and by the neighbor selection strategy. It is more weakly affected by the dimensional strategy.

For $n = 1$, network size is high after uniform neighbor selection and low after weighted neighbor selection.

For $1 < n < \infty$, the converse is true: Network size is high after weighted neighbor selection and low after uniform neighbor selection.

Considering real-world networks, for any fixed choice of n and neighbor selection strategy, the network size produced by one-dimensional revisiting algorithms is lower than that produced by multidimensional independent ones, which in turn is lower than that produced by multidimensional dependent ones. The magnitude of these differences varies by network but is generally lower than the differences between different choices for n and neighbor selection strategy.

This is not the case in Erdős–Rényi and Barabási–Albert, where results across different dimensional strategies are basically the same, and in Barabási–Albert, where some results in the above comparison are actually higher, not lower.

For Erdős–Rényi, network size remains by far the closest to the original size and with the least differences (92.9% – 93.3% of the original size) for all revisiting algorithms except the three algorithms with $1 < n < \infty$ and uniform neighbor selection (74.3% – 75.8% of the original size).

In all real-world networks, there is the same ordering of the four combinations of n and neighbor selection (from highest to lowest resulting network size):

1. $n = 1$ uniform
2. $1 < n < \infty$ weighted
3. $1 < n < \infty$ uniform
4. $n = 1$ weighted[3]

Among the real-world networks, Studentenportal Ilmenau preserves the highest relative network sizes for every configuration. The undirected version of Web of Trust preserves higher relative network sizes than the directed version for every configuration.

See the network sizes for revisiting algorithms for real-world networks illustrated in Figure 6.7.

For tabulated network size data, please refer to Appendix A.

[3] The only exception is the one-dimensional strategy for Studentenportal Ilmenau, where $n = 1$ weighted comes in higher than $1 < n < \infty$ uniform.

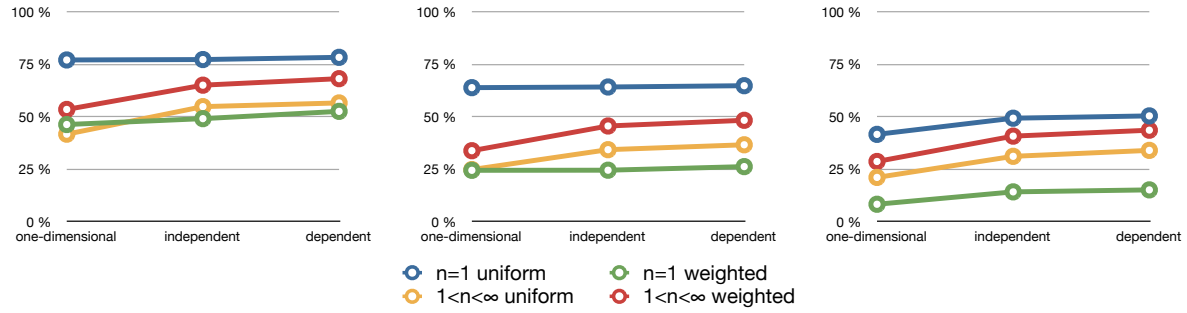

Figure 6.7.: Network sizes produced by revisiting algorithms for SPI, WOT (undirected) and WOT (directed)

6.5.2 Degree Distribution

As expected, BFS retains the highest-degree nodes of all subsampling algorithms, whereas UNS retains the least high-degree nodes, with the degree distribution curves of UNS always shifted substantially to the left compared to all others; moreover, they have by far the largest confidence intervals.

As expected, the degree distributions of $RW_{weighted}$ and $FS_{uniform}$ are each less shifted toward high-degree nodes than that of $RW_{uniform}$; by way of example, the respective out-degree distributions for Studentenportal Ilmenau[4] are depicted in Figure 6.8. Also, the degree distribution of our concept combination $FS_{weighted}$ is even less shifted toward high-degree nodes than that of $RW_{weighted}$ and $FS_{uniform}$.

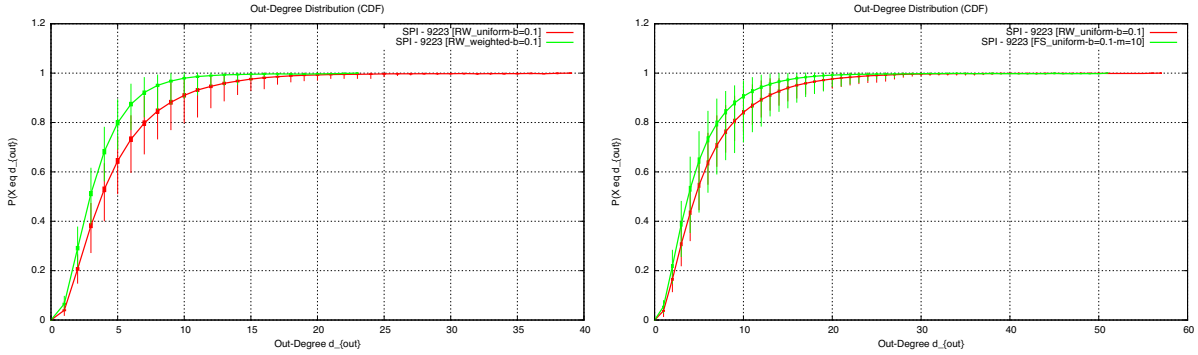

Figure 6.8.: Out-degree distribution produced by $RW_{uniform}$ (red) compared to those by $RW_{weighted}$ and $FS_{uniform}$, respectively (each green), for SPI

Likewise as expected, the degree distribution of our concept combination $SS_{weighted}$ is less shifted toward high-degree nodes than those of $SS_{ordered}$ and $SS_{uniform}$. $SS_{ordered}$ closely matches $SS_{weighted}$, not $SS_{uniform}$.

Similarly, the weighted variants of RW-multi, FS, RDS, RDS-multi, FS+ and FF all produce degree distributions less shifted toward high-degree nodes than their respective uniform variants on most networks.

By contrast, in case of the DFS variants, $DFS_{uniform}$ matches $DFS_{weighted}$, while the degree distribution of $DFS_{ordered}$ is less shifted toward high-degree nodes than those of the other two.

RW-multi produces degree distributions slightly less shifted toward high-degree nodes than RW.

[4] using early termination as described in 6.1.2

6.5.3 Diameter and Characteristic Path Length

The average diameter and characteristic path length turn out to be much affected by the average size of the subsampled networks, rendering comparisons with revisiting subsampling algorithms difficult. The average diameter and characteristic path length produced by non-revisiting algorithms are depicted in Figure 6.9.

In all evaluated undirected networks, UNS produces the highest diameter and characteristic path length of all non-revisiting subsampling algorithms. By contrast, in the one directed network under evaluation, WOT (directed), the two values are among the lowest.

BFS produces the lowest diameter and characteristic path length, except in Barabási–Albert, where it is the second-highest.

The SS variants produce lower diameter and characteristic path length than the DFS variants in all evaluated networks except Erdős–Rényi. Erdős–Rényi, at the same time, is the only set of networks in this evaluation where the diameter and characteristic path length produced by all non-revisiting subsampling algorithms are higher than (or equal to, in the case of BFS) the original.

The following subsampling algorithms produce diameters and characteristic path lengths closest to those of the original networks.

- For Erdős–Rényi: BFS
- For Barabási–Albert: BFS, followed by the DFS variants
- For SPI: the DFS variants
- For WOT (undirected): UNS, followed by the DFS variants
- For WOT (directed): the DFS variants

Overall, the DFS variants retain the diameters and characteristic path lengths closest to the original values.

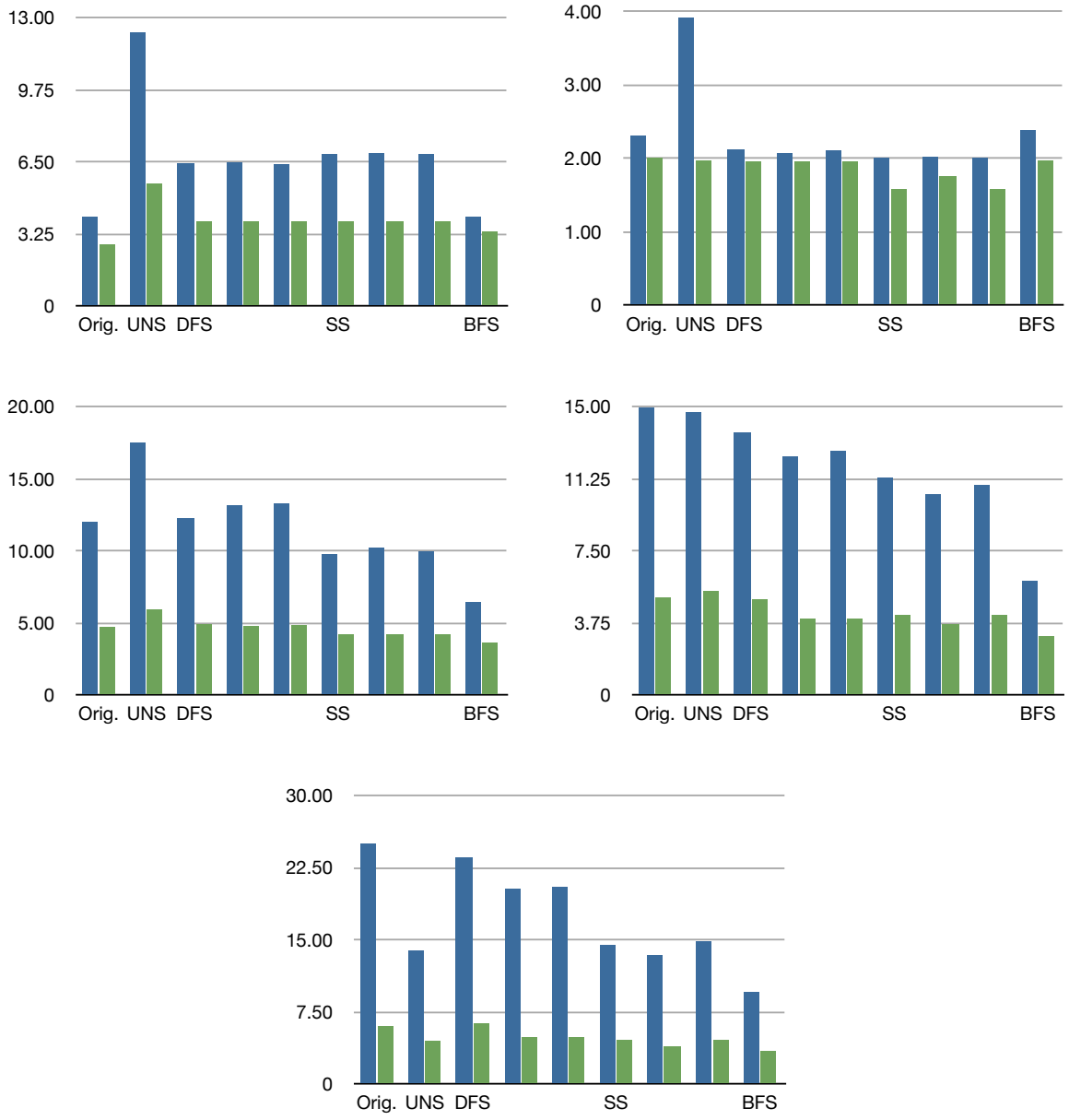

Figure 6.9.: Average diameter and characteristic path length produced by non-revisiting algorithms. From left to right: ER, BA, SPI, WOT (undirected) and WOT (directed)

6.5.4 Clustering Coefficient

The average clustering coefficient in percent of the original network is depicted in Figure 6.10.

In all evaluated networks, UNS produces by far the lowest clustering coefficient (19.6% – 65.9% of the value in the original network). The other subsampling algorithms, with the exception of BFS in Erdős–Rényi, produce results more or less close to the original network (85.2% – 157.9% of the original across all networks, with the values of each network alone spread much less apart). The varying resulting network sizes of different subsampling algorithms appear to make no difference, not even some of the very low ones in Barabási–Albert.

For Erdős–Rényi, the clustering coefficient produced by BFS is extremely high (453.7% of the original), but not for the other networks, where it is still among the highest, but not disproportionally so.

Barabási–Albert is the only evaluated network in which most clustering coefficients after subsampling are lower than before (85.2% – 101.5% of the original). In all other networks, all subsampling algorithms except UNS produce higher clustering coefficients than the original network (109.4% or more of the original).

For the directed WOT, the clustering coefficients of most subsampling algorithms are higher than those of their counterparts for the undirected WOT.

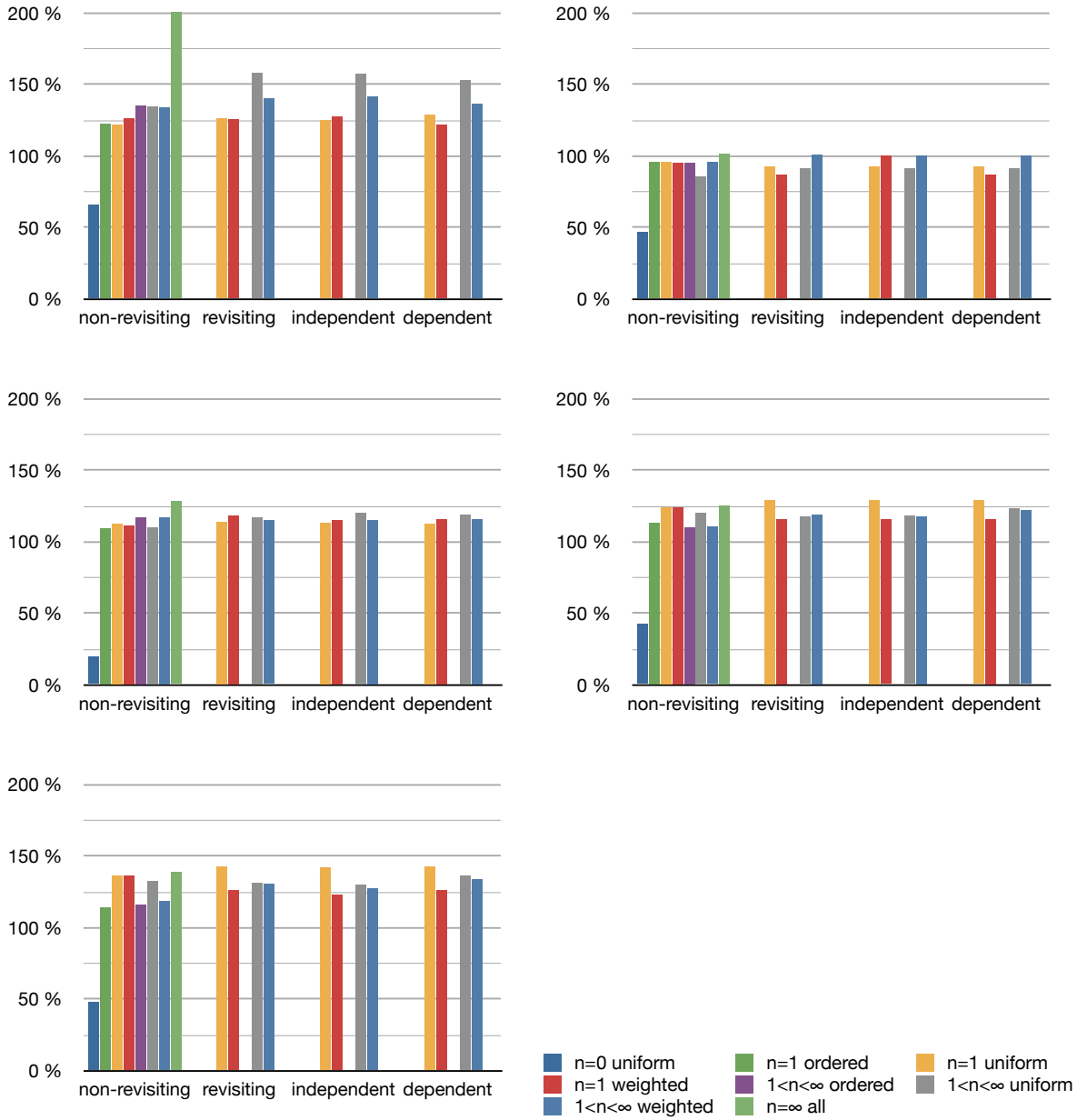

Figure 6.10.: Average clustering coefficient in percent of the original network. From left to right: ER, BA, SPI, WOT (undirected) and WOT (directed)

7 Summary and Conclusion

Our two-dimensional classification by neighbor selection and dimensional strategy has been successful in that we have been able to link certain properties of subsampled networks to categories of subsampling algorithms (as opposed to only individual ones).

We have found out that the resulting network size of revisiting subsampling algorithms is primarily determined by their neighbor selection strategy and secondarily by their dimensional strategy (as well as the ordering within each of these).

We have reproduced the known results for $RW_{uniform}$, $RW_{weighted}$ and $FS_{uniform}$ regarding degree distribution. Our concept combination $FS_{weighted}$ is even less shifted toward high-degree nodes, as expected. We have shown the same effect in the weighted variants of most evaluated groups of subsampling algorithms on most networks.

We have found out that the clustering coefficient is well preserved by all evaluated subsampling algorithms except UNS and BFS, i.e. by subsampling algorithms choosing a limited but nonzero number of neighbors in every step.

We have found the algorithms choosing zero or all neighbors to be subsampling algorithms of extremes. UNS produces by far the lowest node degrees combined with the largest confidence intervals, often the highest diameter and characteristic path length, and by far the lowest clustering coefficient. BFS produces the highest-degree nodes, often the lowest diameter and characteristic path length, and the highest clustering coefficient.

The DFS variants overall produce diameters and characteristic path lengths closest to those of the original networks. $DFS_{ordered}$ often produces a lower clustering coefficient than the other DFS algorithms. $DFS_{uniform}$ and $DFS_{weighted}$ produce near-matching degree distributions.

A Appendix

This appendix contains the average resulting network sizes for all revisiting (both one- and multidimensional) subsampling algorithms and all evaluated networks.

absolute		revisiting	independent	dependent
$n = 1$	uniform	932.52	932.68	932.45
	weighted	930.41	929.25	930.34
$1 < n < \infty$	uniform	743.21	757.80	756.05
	weighted	932.18	931.83	933.49

percent		revisiting	independent	dependent
$n = 1$	uniform	93.3	93.3	93.2
	weighted	93.0	92.9	93.0
$1 < n < \infty$	uniform	74.3	75.8	75.6
	weighted	93.2	93.2	93.3

Table A.1.: Network sizes for Erdős–Rényi (budget $= 1,000$)

absolute		revisiting	independent	dependent
$n = 1$	uniform	541.23	541.98	545.99
	weighted	43.69	49.99	50.92
$1 < n < \infty$	uniform	446.14	465.67	457.10
	weighted	727.09	698.50	696.65

percent		revisiting	independent	dependent
$n = 1$	uniform	54.1	54.2	54.6
	weighted	4.4	5.0	5.1
$1 < n < \infty$	uniform	44.6	46.6	45.7
	weighted	72.7	69.9	69.7

Table A.2.: Network sizes for Barabási–Albert (budget $= 1,000$)

absolute		revisiting	independent	dependent
$n = 1$	uniform	711.60	713.45	723.23
	weighted	427.51	453.36	485.41
$1 < n < \infty$	uniform	385.02	506.27	522.54
	weighted	493.86	600.69	629.33

percent		revisiting	independent	dependent
$n = 1$	uniform	77.2	77.4	78.4
	weighted	46.4	49.2	52.6
$1 < n < \infty$	uniform	41.8	54.9	56.7
	weighted	53.6	65.2	68.3

Table A.3.: Network sizes for Studentenportal Ilmenau (budget $= 922$)

absolute		revisiting	independent	dependent
$n = 1$	uniform	1,629.98	1,637.93	1,653.63
	weighted	624.49	625.86	669.48
$1 < n < \infty$	uniform	633.67	875.55	935.12
	weighted	863.10	1,163.63	1,232.88

percent		revisiting	independent	dependent
$n = 1$	uniform	64.0	64.3	64.9
	weighted	24.5	24.6	26.3
$1 < n < \infty$	uniform	24.9	34.4	36.7
	weighted	33.9	45.7	48.4

Table A.4.: Network sizes for Web of Trust, undirected (budget $= 2,548$)

absolute		revisiting	independent	dependent
$n = 1$	uniform	1,061.73	1,258.21	1,286.38
	weighted	211.78	362.40	386.57
$1 < n < \infty$	uniform	538.08	794.69	867.03
	weighted	731.97	1,040.15	1,112.42

percent		revisiting	independent	dependent
$n = 1$	uniform	41.7	49.4	50.5
	weighted	8.3	14.2	15.2
$1 < n < \infty$	uniform	21.1	31.2	34.0
	weighted	28.7	40.8	43.7

Table A.5.: Network sizes for Web of Trust, directed (budget $= 2,548$)

List of Figures

List of Tables

Bibliography

[DR03] Dimitropoulos, X.A. ; Riley, G.F.: Creating realistic BGP models. In: *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium*, 2003. – ISSN 1526–7539, S. 64 – 70

[GKBM10] Gjoka, M. ; Kurant, M. ; Butts, C.T. ; Markopoulou, A.: Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In: *INFOCOM, 2010 Proceedings IEEE*, 2010. – ISSN 0743–166X, S. 1 –9

[Goo61] Goodman, Leo A.: Snowball Sampling. In: *The Annals of Mathematical Statistics* 32 (1961), Nr. 1, pp. 148-170. http://www.jstor.org/stable/2237615. – ISSN 00034851

[HL71] Holland, Paul W. ; Leinhardt, Samuel: *Transitivity in structural models of small groups*. 1971

[KMT10] Kurant, M. ; Markopoulou, A. ; Thiran, P.: On the bias of BFS (Breadth First Search). In: *Teletraffic Congress (ITC), 2010 22nd International*, 2010, S. 1 –8

[LF06] Leskovec, Jure ; Faloutsos, Christos: Sampling from large graphs. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA : ACM, 2006 (KDD '06). – ISBN 1–59593–339–5, 631–636

[LH07] Leskovec, Jure ; Horvitz, Eric: *Planetary-Scale Views on an Instant-Messaging Network*. 2007

[Lov93] Lovász, L.: *Random Walks on Graphs: A Survey*. 1993

[Mil67] Milgram, Stanley: The Small World Problem. In: *Psychology Today*. 1967, S. 60–67

[MRR+53] Metropolis, Nicholas ; Rosenbluth, Arianna W. ; Rosenbluth, Marshall N. ; Teller, Augusta H. ; Teller, Edward: Equation of State Calculations by Fast Computing Machines. In: *Journal of Chemical Physics* 21 (1953), S. 1087–1092

[RT10] Ribeiro, Bruno ; Towsley, Don: Estimating and sampling graphs with multidimensional random walks. In: *Proceedings of the 10th annual conference on Internet measurement*. New York, NY, USA : ACM, 2010 (IMC '10). – ISBN 978–1–4503–0483–2, 390–403

[SBS+] Schiller, Benjamin ; Bradler, Dirk ; Schweizer, Immanuel ; Mühlhäuser, Max ; Strufe, Thorsten: *GTNA – A Framework for the Graph-Theoretic Network Analysis*

[SR06] Stutzbach, Daniel ; Rejaie, Reza: On Unbiased Sampling for Unstructured Peer-to-Peer Networks. In: *in Proc. ACM IMC*, 2006, S. 27–40

[SRD+06] Stutzbach, D. ; Rejaie, R. ; Duffield, N. ; Sen, S. ; Willinger, W.: Sampling Techniques for Large, Dynamic Graphs. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006. – ISSN 0743–166X, S. 1 –6

[SWM05] Stumpf, Michael P. H. ; Wiuf, Carsten ; May, Robert M.: Subnets of scale-free networks are not scale-free: Sampling properties of networks. In: *Proc. Natl. Acad. Sci. U. S. A.* 102 (2005), Nr. 12, S. 4221–4224. http://dx.doi.org/10.1073/pnas.0501179102. – DOI 10.1073/pnas.0501179102

[WS98] Watts, Duncan J. ; Strogatz, Steven: *Collective dynamics of 'small-world' networks*. 1998