stats

Article

Benford Networks

Roeland de Kok [1] and Giulia Rotundo [2,*]

[1] Land Consult, landConsult.de Öhinghaltweg 3 D, 77815 Bühl, Germany
[2] Department of Statistical Sciences, Sapienza University of Rome, Piazzale Aldo Moro 5, 00185 Roma, Italy
* Correspondence: giulia.rotundo@uniroma1.it

Abstract: The Benford law applied within complex networks is an interesting area of research. This paper proposes a new algorithm for the generation of a Benford network based on priority rank, and further specifies the formal definition. The condition to be taken into account is the probability density of the node degree. In addition to this first algorithm, an iterative algorithm is proposed based on rewiring. Its development requires the introduction of an ad hoc measure for understanding how far an arbitrary network is from a Benford network. The definition is a semi-distance and does not lead to a distance in mathematical terms, instead serving to identify the Benford network as a class. The semi-distance is a function of the network; it is computationally less expensive than the degree of conformity and serves to set a descent condition for the rewiring. The algorithm stops when it meets the condition that either the network is Benford or the maximum number of iterations is reached. The second condition is needed because only a limited set of densities allow for a Benford network. Another important topic is assortativity and the extremes which can be achieved by constraining the network topology; for this reason, we ran simulations on artificial networks and explored further theoretical settings as preliminary work on models of preferential attachment. Based on our extensive analysis, the first proposed algorithm remains the best one from a computational point of view.

Keywords: Benford law; complex networks; semi-distance

Citation: de Kok, R.; Rotundo, G. Benford Networks. *Stats* **2022**, *5*, 934–947. https://doi.org/10.3390/stats5040054

Academic Editors: Claudio Lupi, Roy Cerqueti and Marcel Ausloos

Received: 28 July 2022
Accepted: 20 September 2022
Published: 30 September 2022

1. Introduction

Complex networks showing the properties of Benford's Law can be regarded as Benford Networks (BN). A literature review suggests a gap in the development of this topic. Erdős–Rényi, Watts–Strogatz, and Barabasi–Albert are paradigmatic networks which have been widely used for data modeling [1]. Benford's Law (BL) has been raised to the attention of the public as a tool for fraud detection. Tests on the validity of BL as applied social networks show the applicability of this kind of randomness in networks based on human activities (accounting data, census data, etc.) [2–5].

Yet, there is neither extensive literature on BN nor any definition of the distance to a BN. The generation of artificial BN is not well represented, as the focus to date has been mostly on the application of datasets. This paper aims to fill this research gap by proposing an algorithm for fast and accurate generation of a BN. The speed is due to the method of construction, which foresees the creation of ranks for matching the nodes as extremities of the edges. While the literature contains a few papers proposing the priority ranks for the generation of networks, BNs have not previously been considered in this context [1,6]. Most diffused rewiring algorithms for the creation of target networks have two drawbacks in their computational cost and in the need for a specific network to measure the distance from it. Although it is not computationally efficient, we set up a rewiring algorithm for the creation of BNs in order to explore the edge densities compatible with BNs. This task immediately triggers the need to understand the distance of a network created or rewired from a BN. The existing and well-used conformity degree appears either too rough or too computationally expensive for fine-tuning simulations [7–9]. Measures of distance among networks such as the Hamming, Levenshtein, Jaro–Winkler, and Monge–Elkan ones, to cite a few, have been developed in graph theory, and are used in very different fields from spin glasses to linguistics [10–14]. These can be considered as particular cases of graph edit distances, wherein the base concept is the representation of graphs as strings and the calculus of the number of manipulations needed to go from one graph to the other [15–17]. The main drawback of such distances remains the computational time. The exact computation of such measures is NP-hard, and a reduction of the computational time needs to either approximate solutions or restrict the class of graphs [18]. In the field of complex networks, the difference between two networks has been based mostly on the centrality measures, invariants of the networks, common organizational principles, and more recently on the Laplacian [15,19,20]. Computational time continues to be an issue. As to the second drawback, it is worth pointing out that each of the above-mentioned distances requires two given networks due to the definitions. In fact, the property of being a BN is identifying a set of networks, not a single network. Therefore, the question of measuring how far a network is from a BN is equivalent to asking the distance of the network from a set. This requires determination of the best BN to use, which in turn requires additional computation time. Furthermore, distances among sets, such as the Jaccard index do not improve the computational time, as the best BN network to use as a comparison still has to be determined [21,22].

Therefore, in this study we introduce formal definitions which lead to a semi-metric network space. We compares the computational complexity and show that the proposed rank-based algorithm remains faster than all the other rewiring algorithms.

The rest of this paper is arranged as follows: the next section introduces the formal definitions; Section 3 shows the algorithms; Section 3.1 explains the fast algorithm based on priority ranking; and Section 3.2 outlines a rewiring algorithm, provides an analysis of the assortativity as a function of the density, and discusses the construction of additional algorithms. Last, we elaborate further on the notion of the distance to a BN.

2. Formal Definitions

This section introduces formal definitions. First, we recall that BL describes the probability distribution of the first digit.

Definition 1. *A set of numbers is said to satisfy BL if the leading digit x ($x \in \{1, \ldots, 9\}$) occurs with the following probability distribution:*

$$p(x) = \log_{10}(x+1) - \log_{10}(x) = \log_{10}\left(\frac{x+1}{x}\right) = \log_{10}\left(1 + \frac{1}{x}\right)$$

Table 1 shows the values corresponding to $x \in \{1, \cdots, 9\}$.

Table 1. The distribution of the leading digits in a set following BL.

Leading Digit	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$p(\cdot)$	30.1%	17.6%	12.5%	9.7%	7.9%	6.7%	5.8%	5.1%	4.6%

Remark 1. *The distribution of the leading digits in a set of numbers following BL, properly rounded at the second decimal place and maintaining a sum equal to 1, is provided by*

$$p^{BL} = (0.30, 0.18, 0.12, 0.1, 0.08, 0.07, 0.06, 0.05, 0.04).$$

Focusing on symmetric networks as represented through the adjacency matrix A, where there is an edge among the nodes i and j iff $A(i,j) = A(j,i) = 1$, the degree of a node is a well-known and widely used quantity. Here, we recall its definition for the sake of clarity.

Definition 2. *The degree of a node is the number of its edges.*

Remark 2. *The degree of node i is calculated as $k_i = \sum_j A(i,j) = \sum_j A(j,i)$. The node degree can be considered as a random variable, which leaves room for the definition of a BN.*

Definition 3. *A Benford network (BN) is a network in which the distribution of the leading digit of the node degree follows Benford's Law.*

This definition is in line with the definitions commonly used for Erdős–Rényi (random) and scale-free networks.

For ease of reference, we report the base definitions of assortativity and density commonly used in undirected complex networks [1].

Definition 4. *The assortativity coefficient r of a network is the Pearson correlation coefficient of degree between pairs of nodes connected through an edge.*

Definition 5. *The density of a network is the portion of the potential connections in a network that are actual connections, and is calculated as the ratio of the number of existing edges divided by the total number of potential edges.*

We now need a definition to measure how close a network is to a BN, as well as ensuring that this measure has fast computing time.

After examining the pros and cons of several graph edits, complex networks, and set measures, we found that they suffer from two main drawbacks: long computational time, and the need to identify a specific BN for the calculus. With respect to the first issue, it is worth emphasizing that graph edit distances are NP-hard. A different approach shown in the literature on complex networks is the comparison of their global properties and summary statistics such as network density, degree distribution, transitivity, average shortest path length, and other common organizational principles [20]. However, comparison with a BN only requires checking the node degree; adding other measures does not contribute to determining whether a network is a BN or its distance from a BN. As to the second issue, the problem arises from the fact that the distance measures are based on the presence of two networks. However, the property of being a BN encompasses an entire set of networks, exactly like a scale-free network, a pure random network, or a Watts–Strogatz small world in not identifying one specific network. Furthermore, measuring distances among sets, for example using the Jaccard index, incurs the same problem. Thus, in order to calculate the distance through this approach a single BN should be selected, which adds a further optimization problem to solve. Therefore, we follow another approach here, focusing only on the characterization of BNs through BL.

Definition 6. *Given two networks, A and B, we define $d(A,B)$ as the distance among the histograms of the leading digits of the node degrees.*

Because we need to estimate the frequencies of the leading digit, $d(A,B)$ is actually a distance among vectors in R^n, where $n = 9$. Hence, we base it on the sum of the absolute values of the differences (i.e., norm 1 of the difference among the vectors), although any equivalent distance can be used.

Definition 7. *Let $p^A = (p_1^A, \cdots, p_n^A)$ and $p^B = (p_1^B, \cdots, b_n^B)$ be the set of the y-values of the histogram (i.e., the frequency) of the leading digit of the node degree calculated on the networks A and B, respectively. Then, the distance $d(A,B)$ among the two networks A and B is $d(A,B) = \sum_{i=1}^n \mid p_i^A - p_i^B \mid$.*

Remark 3. *If a network, let us say B, is a BN, then $p^B = p^{BL}$.*

Definition 8. *The distance $d_{BN}(A) = d(A,BN)$ of an empirical network A from a BN is $d(A,BN) = \sum_{i=1}^n \mid p_i^A - p_i^{BL} \mid$, that is, the sum of the absolute values of the differences among the p^{BL} and the y-values of distribution of the leading digit of the node degrees of the network A. This definition applied to a raw vector is not far from the Mean Absolute Distance (MAD) [9], which is its average; however, in this paper it is used for characterizing a network, using the node degree as an intermediate step. In this setting, $d_{BN}(\cdot) = d(\cdot, BN)$ is merely a particular case of $d(\cdot,\cdot)$.*

Definition 9. *A network A is considered to be a BN when* $d_{BN}(A) = 0$.

In order to show the application of this distance, we examine several real-world data sets retrieved from the Stanford Large Network Dataset Collection repository (SNAP) for scientific collaboration networks and Facebook [23]. These collaboration networks consist of data extracted from the ArXiv sections on Astrophysics (AstroPh), Condensed Matter (CondMat), General Relativity (GrQc), High Energy Physics (HepPh), and High Energy Physics Theory (HepTh). If an author i co-authored a paper with author j, the graph contains an undirected edge connecting i and j [24]. The Facebook dataset consists of anonymized data collected from survey participants using a specific Facebook app [25]. Figure 1 shows the histogram of the leading digit of the node degrees.

Table 2 reports the number of nodes, the number of edges, and $d_{BN}(\cdot) = d(\cdot, BN)$.

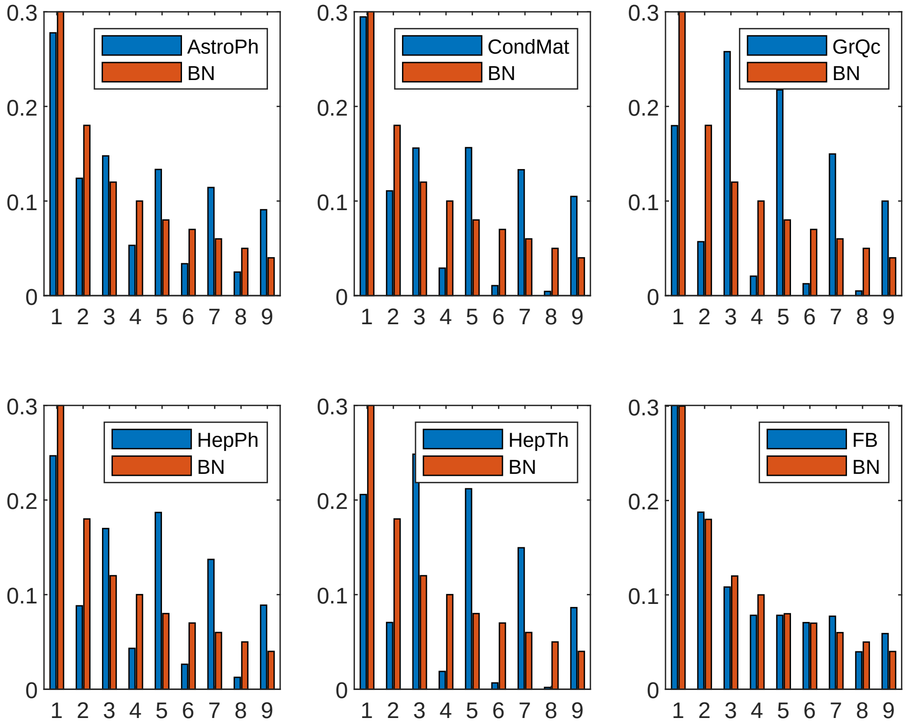

Figure 1. Example of the histogram of the node degree of datasets from SNAP (collaboration networks from ArXiv (Astrophysics (AstroPh), Condensed Matter (CondMat), General Relativity (GrQc), High Energy Physics (HepPh), High Energy Physics Theory (HepTh)) and from Facebook).

Table 2. Analysis of the datasets: the first columns report the description and number of edges, while the last column shows the distance $d(\cdot, BN)$.

Description	Nodes	Edges	$d(\cdot, BN)$
Astro Physics	18,772	198,110	0.3725
Condensed Matter	23,133	93,497	0.5009
General Relativity	5242	14,496	0.8502
High Energy Physics	12,008	118,521	0.5657
High Energy Physics Theory	9877	25,998	0.7923
Facebook 2	1034	54015	0.0907

There are quite a few differences among the datasets. In all the collaboration networks there is a high excess of nodes in bin 5 compared to the BN, meaning that the papers are quite frequently coauthored by groups. This is maximal in General Relativity, where the distance from a BN is higher than in the other datasets. The distribution of the first digit in the Condensed Matter section is quite close to the BN. Because the presence of one edge implies co-authorship among two nodes and each node represents an author, the histogram emphasizes the prevailing amount of papers co-authored by two scientists. However, this is not sufficient to state that the collaborations in Condensed Matter result in the network closest to the BN, as the Astrophysics community has a smaller distance despite not yet being a BN. Definitively, co-authorships cannot be considered to occur at random. The dataset from Facebook is quite different from the others, and shows a network that is much closer to the BN.

When we consider the space of all the networks, $d(A,B)$ a is not a distance in the mathematical sense. In fact, the first two conditions surely hold for the distance ($d(A,A) = 0$, and $d(A,B) = d(B,A)$). The triangular inequality cannot be defined because the sum of two networks is not defined.

Now, we recall the definition of semi-metric space [26].

Definition 10. *Let Z be a set of elements common to each pair that corresponds to a positive real number, which we call the distance between them. If a and b are any two elements, we designate this distance by $d(a,b)$, and can postulate that the following axioms are satisfied: I. $d(a,b) = d(b,a)$; and II. $d(a,b) = 0$, if and only if $a = b$. A space that satisfies these conditions is a semi-metric space.*

Remark 4. *The set of networks together with $d(A,B)$ is a semi-metric space.*

Remark 5. *$d(A,B)$ provides a partial order of the set of network.*

Such networks can be quite different among themselves, as the request on the BN is only on the marginal distribution of the node degree.

3. Algorithms for Simulating BNs

The relevance of introducing a way to detect how far an arbitrary network is from a BN allows us to introduce algorithms for generating a Benford network. To provide practical simulations, we ran the algorithms on a network of 100 nodes, such that the network is a BN when it has 30 nodes with the leading digit of its degree equal to 1, 18 with leading digit 2, etc., as summarized in Table 3.

Table 3. The distribution of the leading digits in a set following BL.

Leading Digit	1	2	3	4	5	6	7	8	9
number of nodes	30	18	12	10	8	7	6	5	4

The conformity tests already in use are not suitable for either accuracy or computational time. For instance, the four levels of conformity proposed in [9], namely, 'close conformity', 'acceptable conformity', 'marginal conformity', and 'nonconformity', are too rough to fine-tune an optimization algorithm. In general, tests of conformity [7,8] are computationally more expensive than the calculus of $d_{BN}(A)$.

In the following subsections, we first introduce a very fast algorithm for generating a BN, then tackle the problem of the BL appearing as a function of the density of the network.

3.1. A Fast Algorithm for a BN with Maximal/Minimal Assortativity

Creating a BN is a first step that can serve as a basis for comparing and testing algorithms. The selection of $N = 100$ nodes is without loss of generality, as for a different number of nodes all that is needed is to recalculate the total number of nodes which contribute to the total count of each leading digit. The overall approach remains the same. Here, we propose an algorithm that immediately builds a BN. The pseudo-code is as follows:

```
1. initialize a network with N nodes and 0 edges
2. assign each node its degree so as to fullfill the BL
3. Unil each degree is reached:
select the beginning and end of each edge
```

which, from the point of view of the adjacency matrix, reads as follows:

```
1. create an NxN matrix A with each element equal to 0
2. create a vector v of length N storing the degree of each node
3. Until each degree is reached:
select i, j, and set A(i,j)=A(j,i)=1
```

The first step is $O(N^2)$, as it involves the creation of a matrix in which each element is equal to 0. Practically, in the second step a list is created in which each node is assigned the desired node degree (for instance, nodes 1–4 are assigned the node-degree 9, nodes 5–9 are assigned the node-degree 8, etc., until the last 30 nodes with degree 1, although this is not the only possibility). This step is $O(N)$, as it consists of reading a vector with N entries.

The third step is the selection of the beginning and end of each edge. To perform the task, the list is scrolled to select the match, which in principle can be done randomly. However, random matching of the beginning and end of each link is not as fast as following a precise criterion, since it involves a pseudo-random number generator. We propose two criteria, one aiming at maximal assortativity, the other at minimal assortativity. Therefore, the last part can be detailed as follows:

```
1. create an NxN matrix A with each element equal to 0
2. create a vector v of length N
   assigning the degree in descending order
3. for each node i=1,\ldots,N
   until its node degree v(i) is reached:
   match the other end j of each edge
   with the first available node
```

in the above, ‘available’ stands for ‘not already connected’, that is, for which the node degree has not already been reached.

Because the order of the degrees is descending, the algorithm begins with the nodes with the highest degree.

The algorithm provides a BN. This is trivial due to the condition of the node degree. Figure 2 shows the network.

Remark 6. *A network obtained in this way gives rise to the maximal assortativity. The condition of the descending order ensures that nodes with a high degree first have edges with nodes with a high degree, and have edges with nodes with a lower degree only when there is no better possibility [1]. Because assortativity is the correlation among the node degrees, any inversion in the sequence immediately decreases the values in the formula.*

The complexity of this match is the same as a roll of the list, assigning the node degree ($O(N)$) and then the second one to find the first available node; thus, with N nodes the complexity is $O(N * (N-1)) \sim O(N^2)$, which is much faster than any other random rewiring procedure, as it avoids the computational time needed for the pseudo-random generator.

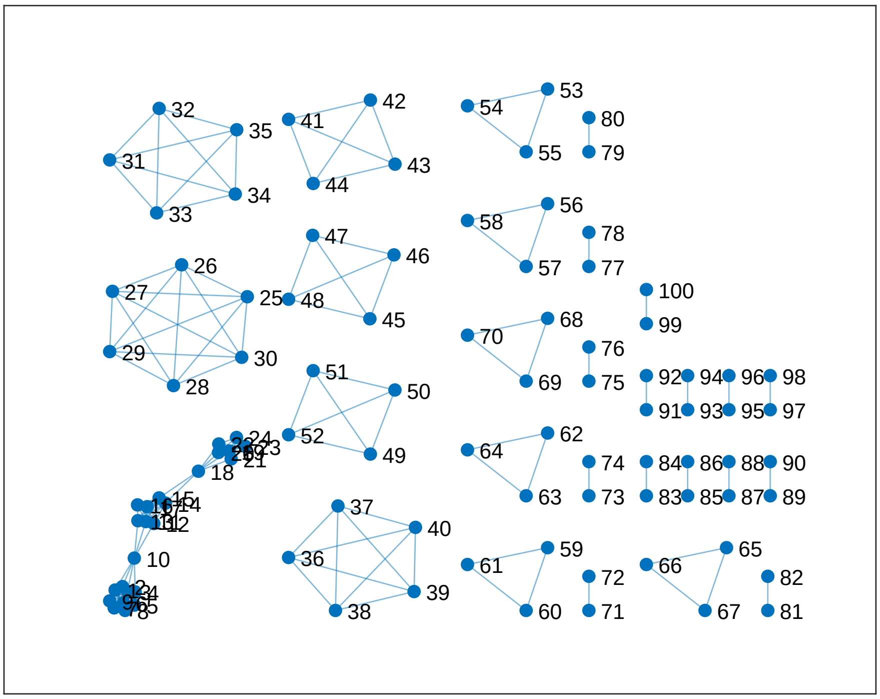

Figure 2. Maximal assortative network provided by our proposed fast algorithm. Nodes with a similar degree tend to be connected among themselves. In the figure, this is very evident in the group of 1-connected units (nodes ranging from 71 to 100), in the group of 2-connected units (nodes ranging from 53 to 70), in the group of 3-connected units (nodes ranging from 41 to 52), and in the group of 4-connected units (nodes ranging from 31 to 40).

Remark 7. *The proposed algorithm has a computational time* $O(N^2)$.

Remark 8. *Here, we introduce a condition to avoid loops (i.e.,* $A(i,i) = 0\ \forall i = 1, \cdots, N$*) except where strictly necessary to match the degree list. In fact, general speaking, not all assignments of degrees to the nodes are compatible with the topology of a network. Figure 3 shows this issue. If loops are not allowed and four nodes have degree four, then the fifth node needs to have degree four as well. For instance, if we assign degree* 3 *to the node, we need to remove one link; hence one of the other nodes, say node b, needs to have its degree decreased to* 3 *as well. Therefore, the set of degrees* $q = (3, 4, 4, 4, 4)$ *is incompatible with the network unless we allow loops.*

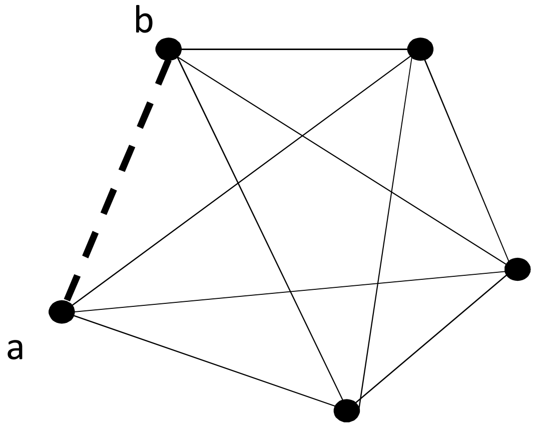

Figure 3. Example of the constraint on the number of edges: if node a has four edges, then node b must have four edges. Node a cannot have three edges if all the other nodes have four edges.

When running the algorithm on a network of 100 nodes, 171 edges are created, which corresponds to $N_{edges} = \frac{1}{2} < p, q >$, where $p = (30, 18, 12, 10, 8, 7, 6, 5, 4)$ and $q = (1, 2, 3, 4, 5, 6, 7, 8, 9)$. The constant $\frac{1}{2}$ is needed due to the bidirectional role of the edges. The density is 0.034, and there are no loops.

The condition on the match among nodes with the closer (higher) degree can be inverted, setting the connections among the nodes with either the highest node degree or the lowest one. The result continues to be a BN network, as the requirements on the BN are unchanged, except now with the assortativity slightly negative and very close to 0, with one link less than needed, resulting in the need to add one loop. The density remains the same. The computational complexity remains the same as well, as the list is simply scrolled in the reverse direction. Figure 4 shows the network.

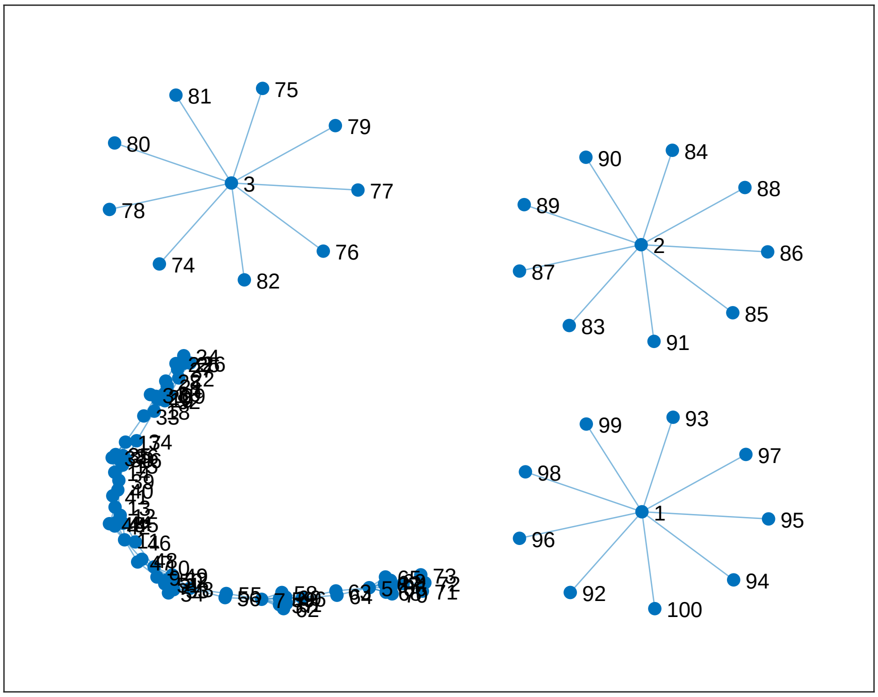

Figure 4. Minimal assortative network provided by the fast algorithm. Nodes 1, 2, and 3 have degree 9 and are connected to a total of 27 of the 1-connected units. There are a total of 30 units with only 1 connection, meaning that node 4, which is the last to have degree 9, can connect to the last three 1-connected units and be connected to the other six nodes of the group of 2-connected nodes. Therefore, the figure does not show a star for node 4. The next group of nodes (from 5 to 9) have eight connections and are linked first to the 2-connected nodes, then to the 3-connected,. . . , etc.

Remark 9. *This is not the only way to create a BN. For instance, in a BN, a node with degree* 1 *makes the same contribution to the distribution as a node with degree* 10, 11, $\cdots$, 19*, as the leading digit remains* 1*. In general, a node contributes to the count of a leading digit x if it has x, x*0*, x*1*,* $\cdots$*, x*9 *edges, meaning that each node degree may have* 11 *different values and contribute to the counting of the same leading digit. The computational complexity remains the same as the function of N, as N* $-$ 1 *is an upper limit for the edges departing from every single node. However, keeping the node degrees as low as possible contributes to the speed of the algorithm (obviously, creating* 30 *connections for the set of the* 30 *nodes with degree* 1 *is* 10 *times faster than creating* 30 × 10 *connections in which each node with degree* 1 *is replaced by a node with degree* 10*).*

3.2. The BN as a Function of the Density of the Network

This section recalls the first results on random networks, where the task was to understand the density required for particular properties. The rationale behind the fact that many densities can be compatible with the validity of the BL on the node degree distribution relies on the fact that only the leading digit contributes to the BL. The same argument as in Remark 9 allows us to calculate the total number of the BN which can be obtained from a network with N nodes. If the identity of each node has to be kept the same, then there are 11^N BN networks (11 possible values for each of the N nodes, where each value can be taken independently from the values of the other nodes). The number of possible networks is simply too high for an exaustive analysis. If we focus on network topology, the identification number of each node is not relevant. For instance, in a group of four nodes with the leading digit of the degree equal to 9, it is not relevant if the first has degree 9 and the remaining three have degree 99, or if the second has degree 9 and the others have degree 99. What matters is how many have degree 9, 91, 92, $\cdots$, 99. Therefore, in each set of nodes having the same leading digit, the number of possible assignments for the node degree is calculated as the number of combinations with repetition of 11 objects. Two combinations with repetition are considered identical if they have the same elements repeated the same number of times, regardless of their order. Recall that the number of combinations of $r = 11$ elements taken at k at each time is

$$\begin{pmatrix} \mathrm{r+k-1} \\ \mathrm{k} \end{pmatrix} \tag{1}$$

Therefore, the total number of networks with topologies different from each other is $P = \prod_{i=1}^{9} \begin{pmatrix} 11+\mathrm{p(i)}-1 \\ \mathrm{p(i)} \end{pmatrix} = 2.7225 \times 10^{46}$ different configurations, where $p = (30, 18, 12, 10, 8, 7, 6, 5, 4)$. As this number of networks remains too high for exaustive generation and analysis of each, we fix a discrete set of densities.

In this section, we first perform a preliminary analysis of the range of densities of BNs, then obtain a picture of the assortativity as a function of the densities through a rewiring procedure.

3.2.1. Analysis of the Range of Densities of BNs

Keeping $N = 100$ as our reference, a BN network in which each node has at least one link and with the minimum number of edges (that is, minimum density) is the same as the one built in the previous section. In fact, the set of node degrees is the lowest which can fit BL. Eventual lower densities of a BN can be obtained if nodes have 0 connections, allowing the percentage of the node degree to fit BL despite being calculated on a lower number of nodes. Alternatively, if we want to increase the number of edges, the minimal amount which we have to add is 9 to move from a node with degree 1 to one with degree 10. This results in a gap in the possible set of densities, while after this value there can be many BNs with intermediate values for the densities, up to the one with the maximum number of edges. The latter has 30 nodes with degree 19, 18 nodes with degree 29, $\cdots$, and 4 nodes with degree 99, due to the role of the leading digit. The number of edges is 2160, which corresponds to $\frac{1}{2} < p, q >$, where $p = (30, 18, 12, 10, 8, 7, 6, 5, 4)$ and $q = (19, 29, 39, 49, 59, 69, 79, 89, 99)$. The density is 0.436.

3.2.2. Rewiring Algorithm

Rewiring is a quite immediate method for achieving a target topology. The pseudo-code can be outlined as follows:

```
1. start from a random seed network with the due density
2. while the network is not a BN
   (or the maximal number of trial is reached)
   2.a select a link for the rewire
   2.b if the rewire produces a network closer to a BN:
        then accept the rewire
        otherwise skip
   end
```

```
3. store the distances from a BN
4. report the data in a figure
```

Remark 10. $d_{BN}(\cdot)$ *is essential for measuring whether the resulting network is closer to a BN, and thus whether to accept the rewire.*

Remark 11. *The algorithm follows a descending direction (i.e., the rewire is accepted only if the distance from a BN decreases).*

Conformity tests can provide an answer regarding either the rejection or acceptance of a probability distribution; this answer need not be only a yes/no, and various scales of conformity degrees can be used [7,9,27]. However, conformity tests are not the best choice for running simulations. Those that provide only four degrees of acceptance ('conformity', ...'not conformity') are too rough to form a basis for simulations. Moreover, it is easy to find through cross-checking that all the conformity tests are computationally more expensive than calculating a histogram and the distance from a vector with 9 components. Our notion of distance does not aim at providing a conformity test, although it is possible to use them to elaborate on the matter as soon as bounds are defined.

Here, we deepen our analysis by focusing on the assortativity. The starting point is a BN; the rewiring aims at either increasing or decreasing the assortativity while maintaining the BN and allowing swapping of the edges. The algorithm is an iterative one, and can be outlined as follows:

```
1. select two links of a BN network
2. if the swap increases (decreases) the assortativity,
   then accept the swap
```

Table 4 summarizes the results of 100, 000 simulation steps. The first row reports the densities which were examined, taken with a step equal to 0.01 below the density 0.1, to obtain fine detail, with step 0.1 being above 0.1. The BN appears from density 0.034 until to density 0.436, shown in bold. Below the minimal density, a BN can still be found if the histogram is calculated on the nodes which have at least one edge. Figure 4 shows a graphical representation of the results of Table 4.

Table 4. Densities calculated as percentages of the total number of links used to run the simulations on rewiring to achieve a BN. The first percentages differ by only 0.01 in order to fine-tune the threshold of the BN. The values above 0.1 differ by 0.1 because the increase in distance from a BN follows relatively stable path. The minimal, maximal, and average distance from a BN are shown, and correspond to the plot in Figure 5.

density	0.01	0.02	0.03	**0.034**	0.04	0.05	0.06	0.07	0.08	0.09
mean ass.	0	0	0.013	0.03	0.05	0.045	0.004	−0.011	−0.016	0.151
min ass.	−0.167	0.006	0.006	0.00	−0.117	−0.105	−0.179	−0.052	0.00	0.00
max ass.	0	0.027	0.191	0.070	0.36	0.385	0.215	0.484	0.314	0.289
density	0.1	0.2	0.3	0.4	**0.436**					
mean ass.	−0.087	0.002	−0.022	0.003	0.00					
min ass.	−0.118	−0.05	−0.147	−0.210	−0.085					
max ass.	0.136	0.191	0.162	0.171	0.020					

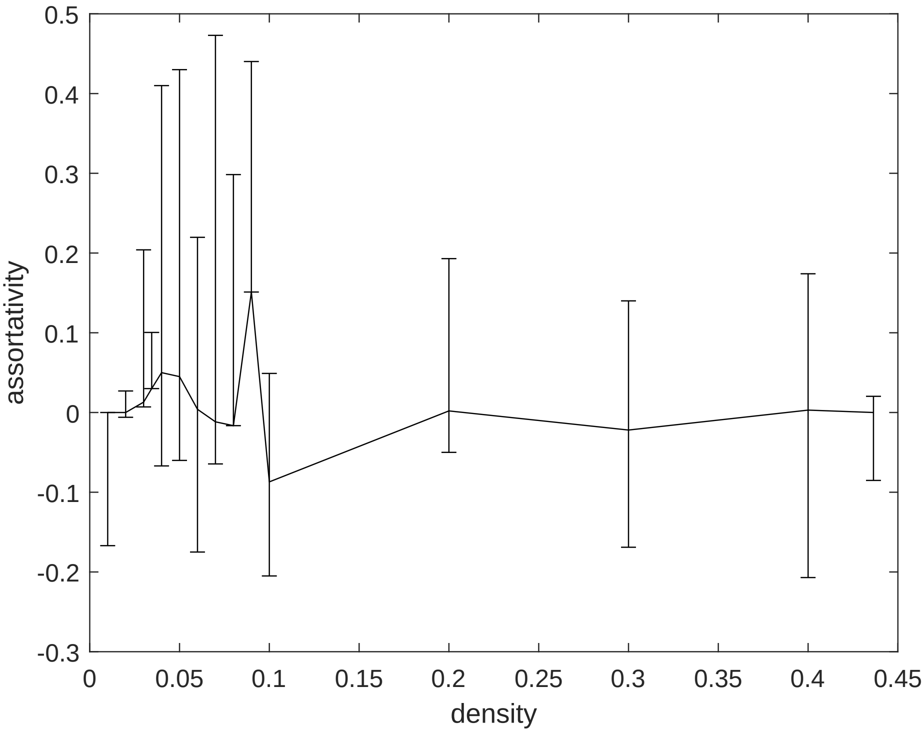

Figure 5. Figure corresponding to Table 4. The mean assortativity is shown as a function of the density. The error bars show the distance between the minimal and maximal assortativity.

3.2.3. An Intermediate Algorithm for the Immediate Construction of a BN and Random Rewiring

The distance from a BN which we use for simulation is fast and accurate. It does not involve a rewiring process, which is computationally more expensive. Suppose, however, that a seed network is assigned as a starting point for simulations. Is it possible to drive the rewiring without random selection of the nodes to be checked? In other words, can the edges to be rewired be selected through targeted distribution? The answer to this question provides a way of targeting the rewiring process. To outline this through an example, we refer to Figure 1, specifically the High Energy Physics collaboration network. The maximal distance from the BL is in the bin corresponding to the leading digit, 5. Removing edges from that set of nodes would quickly improve the proximity of the distribution of the node degrees to the BL. Of course, this targeted selection can be carried out using the distance already introduced by working on the nodes of each bin instead, than selecting them at random. The computational time is $O(N^3)$, as it involves a double reading of a list to determine which nodes need to have other nodes removed or added, followed by another scrolling of the list of nodes to find the match.

4. A New Definition of the Distance to a BN

We now focus on a refinement of the notion of distance. We can formalize the problem as follows: consider x random variables describing the node degree. Of course, it is going to follow a distribution. The question is then which distribution should follow another random variable y such that $z = x + y$ follows a BL. In a formal setting, $p(z) \sim f_{BL}$. In other words, when a random rewire is performed instead of a priority list, how should this random selection be carried out?

Definition 11. *Let X be a random variable; then, a complement to BL is a density such that*

$$f_{X+Y}(a) = f_{BL} \tag{2}$$

Practically speaking, the complement is the perturbation that has to be added to the node degrees to fit BL.

Let us focus on Y independent on X. It well known that

$$f_{X+Y}(a) = \frac{\mathrm{d}}{\mathrm{d}a}\int_{-\infty}^{\infty} F_X(a-y)f_Y(y)dy = \int_{-\infty}^{\infty} \frac{\mathrm{d}}{\mathrm{d}a}F_X(a-y)f_Y(y)dy = \int_{-\infty}^{\infty} f_X(a-y)f_Y(y)dy,$$

hence, (2) becomes $\int_{-\infty}^{\infty} f_X(a-y)f_Y(y)dy = f_{BL}$. In a discrete setting, let p (q) be the vector of discrete probabilities of X (Y) and let $p, q, p_{BL} \in R^n$; then, Condition (2) reads $p^i \cdot q^{n-i} = p^i_{BL}$.

Remark 12. *In a general setting, the complement $\tilde{f}(X) = f_Y(y)$ can be calculated by solving the implicit relation $f_{X+Y}(a) = f_{BL}$.*

Whether specific results for the sum of two random variables are already available, the explicit probability of Y can be detected. This is the case for scale-free networks, which are usually meant for a node degree–exponent power law with exponent $2 < \gamma < 3$, while BL is a power law with exponent 1. Because the sum of two power-law variables is a power law with an exponent the tail of which is dominated by the contribution of the term with the smallest exponent [28,29], when determining whether the starting network is a scale-free one the perturbation of the node degree X for achieving a BN is a random variable Y with exponent 1.

Remark 13. *The complement can be used to introduce a partial order on X.*

The complement is a function, and several measures can be considered for setting a partial order on a set of functions.

Remark 14. *Moreover, the set of class networks together with a partial order on $\tilde{f}(X)$ is a semi-metric space.*

This follows from the properties of the partial order on the functions and free gathering of the networks into sets. For instance, while two networks may be not identical, if they are both BNs they belong to the same class.

Remark 15. *The fast and accurate algorithm presented in Section 3 remains the fastest for generating a BN, as random rewiring, though targeted, requires additional computational time for random variable generation and eventual acceptance or rejection of the rewire.*

5. Discussion and Conclusions

This paper is based on the notion of a gap existing in the literature concerning the application of BL to complex networks. We introduce a clear definition of a BN. Our main aim in this paper is to provide elements for BN simulation settings.

The first algorithm, which we propose in Section 3.1, is a priority-rank based algorithm. It is fast and accurate, and is based on the creation of a match-list for assigning the edges. This choice is faster than any random assignment, which involves the added computational time of a random generator and the eventual rejection of selections, leading to even more computational steps. The availability of a fast algorithm is a key element for further studies on both properties of BNs and comparison with real-world datasets. We have proven that this algorithm is the best computational choice in comparison to random rewiring procedures, which in turn require the development of a way to measure the distance between an arbitrary network and a BN. Defining distance among networks for the purpose of measuring how close a network is to a BN is not trivial. Our examination of commonly used measures of distance among networks indicated that they are not effective for measuring distance from a BN, mainly due to high computational times [15,19–22]. For instance, the networks shown in Figures 2 and 4 are BNs, although with quite different topologies. The definition cannot be substituted by the conformity degrees due to their lack of precision. Defining the distance is the first step in setting up algorithms to generate BNs. Therefore, in order to compare our algorithm with a random rewiring procedure, we introduce a new semi-measure of the distance of a network from a BN and present an analysis of the assortativity as a function of the density of the network. The last part proposes a theoretical approach which opens the way for further exploration of mechanisms of preferential attachment.

In summary, we trust that the results shown in this paper will add insights regarding BNs and serve as the basis for future work and development.

Author Contributions: All authors equally contributed to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors thank COST Action CA18232—Mathematical models for interacting dynamics on networks (MAT-DYN-NET) for fruitful scientific talks and further interaction at the workshop held in Rome on 14–16 September 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Barabási, A.L. *Network Science*; Cambridge University Press: Cambridge, UK, 2016.
2. Ausloos, M.; Herteliu, C.; Ileanu, B. Breakdown of Benford's law for birth data. *Phys. A Stat. Mech. Appl.* **2015**, *419*, 736–745. [CrossRef]
3. Belluzzo, T. Benford's Law. GitHub. 2022. Available online: https://github.com/TommasoBelluzzo/BenfordLaw (accessed on 31 May 2022).
4. Hassler, U.; Hosseinkouchack, M. Testing the Newcomb-Benford Law: Experimental evidence. *Appl. Econ. Lett.* **2019**, *26*, 1762–1769. [CrossRef]
5. Morzy, M.; Kajdanowicz, T.; Szymański, B.K. Benford's Distribution in Complex Networks. *Sci. Rep.* **2016**, *6*, 34917. [CrossRef] [PubMed]
6. Morzy, M.; Kazienko, P.; Kajdanowicz, T. Priority rank model for social network generation. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, USA, 18–21 August 2016; pp. 315–318.
7. Cerqueti, R.; Lupi, C. Some New Tests of Conformity with Benford's Law. *Stats* **2021**, *4*, 745–761. [CrossRef]
8. Cerqueti, R.; Maggi, M. Data validity and statistical conformity with Benford's Law. *Chaos Solitons Fractals* **2021**, *144*, 110740. [CrossRef]
9. Nigrini, M.J. Benford's Law: Assessing Conformity. In *Forensic Analytics: Methods and Techniques for Forensic Accounting Investigations*; 2012. Available online: https://onlinelibrary.wiley.com/doi/10.1002/9781118386798.ch6 (accessed on 1 June 2022).
10. Angeles, M.; Espino-Gamez, A. Comparison of methods Hamming Distance, Jaro, and Monge-Elkan. In Proceedings of the International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA 2015, Rome, Italy, 24–29 May 2015.
11. Chaabi, Y.; Allah, F. Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *34*, 6116–6124. [CrossRef]
12. Jimenez, S.; Becerra, C.; Gelbukh, A.; Gonzalez, F. Generalized Mongue-Elkan Method for Approximate Text String Comparison. In *Computational Linguistics and Intelligent Text Processing. CICLing 2009*; Gelbukh, A., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5449.
13. Kashefi, O.; Sharifi, M.; Minaie, B. A novel string distance metric for ranking Persian respelling suggestions. *Nat. Lang. Eng.* **2013**, *19*, 259–284. [CrossRef]
14. Nishimura, K.; Nishimori, H.; Ochoa, A.J.; Katzgraber, H.G. Retrieving the ground state of spin glasses using thermal noise: Performance of quantum annealing at finite temperatures. *Phys. Rev. E* **2016**, *94*, 032105. [CrossRef]
15. Emmert-Streib, F.; Matthias Dehmer, M.; Shi, Y. Fifty years of graph matching, network alignment and network comparison. *Inf. Sci.* **2016** *346–347*, 180–197. [CrossRef]
16. Gao, X.; Xiao, B.; Tao, D.; Li, X. A survey of graph edit distance. *Pattern Anal. Appl.* **2010**, *13*, 113–129. [CrossRef]
17. Li, T.; Dong, H.; Shi, Y.; Dehmer, M. A comparative analysis of new graph distance measures and graph edit distance. *Inf. Sci.* **2017**, *403–404*, 15–21. [CrossRef]
18. Bougleuxa, S.; Bruna, L.; Carletti, V.; Foggia, P.; Gaüzére, B.; Vento, M. Graph Edit Distance as a Quadratic Assignment Problem. *Pattern Recognit. Lett.* **2016**, *87*, 38–46. [CrossRef]
19. Shimada, Y.; Hirata, Y.; Ikeguchi, T.; Aihara, K. Graph distance for complex networks. *Sci. Rep.* **2016**, *6*, 34944. [CrossRef] [PubMed]

20. Wegner, A.; Ospina-Forero, L.; Gaunt, R.; Deane, C.; Reinert, G. Identifying networks with common organizational principles. *J. Complex Netw.* **2018**, *6*, 887–913. [CrossRef]
21. Jaccard, P. The Distribution of the Flora in the Alpine Zone.1. *New Phytol.* **1912**, *11*, 37–50. [CrossRef]
22. Kosub, S. A note on the triangle inequality for the Jaccard distance. *Pattern Recognit. Lett.* **2019**, *120*, 36–38. [CrossRef]
23. Leskovec, J. Stanford Large Network Dataset Collection Repository. Available online: https://snap.stanford.edu/data/index.html#citnets (accessed on 1 June 2022).
24. Leskovec, J.; Kleinberg, J.; Faloutsos, C. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 2-es. [CrossRef]
25. McAuley, J.; Leskovec, J. Learning to Discover Social Circles in Ego Networks. In Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012.
26. Wilson, A.W. On Semi-Metric Spaces. *Am. J. Math.* **1931**, *53*, 361–373. [CrossRef]
27. Holst, E.; Thyregod, P.; Wilrich, P. On Conformity Testing and the Use of Two Stage Procedures. *Int. Stat. Rev. Int. Stat.* **2001**, *69*, 419–432. [CrossRef]
28. Arnold, B.C. Pareto Distributions. In *Monographs on Statistics and Applied Probability*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2015; Volume 140.
29. Wilke, C.; Altmeyer, S.; Martinetz, T. Large-scale evolution and extinction in a hierarchically structured environment. *arXiv* **1998**, arXiv:adap-org/9803001.