Percolation on sparse networks

Brian Karrer,[1] M. E. J. Newman,[1] and Lenka Zdeborová[2]

[1]*Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA*
[2]*Institut de Physique Théorique, CEA Saclay and URA 2306, CNRS, Gif-sur-Yvette, France*

We study percolation on networks, which is widely used as a model of the resilience of networked systems such as the Internet to attack or failure and as a simple model of the spread of disease over human contact networks. We reformulate percolation as a message passing process and use the resulting equations to show, among other things, that for sparse networks, which includes most networks observed in the real world, the percolation threshold is given by the inverse of the leading eigenvalue of the so-called non-backtracking matrix. Like most message passing calculations, our results are exact on networks that have few small loops but, as we show, they also provide bounds on the percolation behavior of networks that do contain loops.

Percolation, the random occupation of sites or bonds on a lattice or network, is one of the best-studied processes in statistical physics. It is used as a model of porous media [1, 2], granular and composite materials [3–6], resistor networks [7], forest fires [8], polymers [9], and many other systems of scientific interest. In this paper we study the bond (or edge) percolation process on general networks or graphs, which is used to model the spread of disease [10, 11] and network robustness [12–14] in social and technological networks, among other things. Although percolation has been studied extensively on simple model networks such as random graphs, there are few analytic results for real-world networks, whose structure is typically more complicated. One important and powerful result has been demonstrated recently by Bollobás *et al.* [15], who show that in the limit of large network size the threshold occupation probability for bond percolation on a dense network is equal to the reciprocal of the leading (most positive) eigenvalue of the adjacency matrix. "Dense" in this case means that a nonzero fraction of all possible edges are present in the network in the limit of large size, or equivalently that the number of edges increases as n^2 in a network of n nodes. Unfortunately, few real-world networks satisfy this requirement. Most real-world networks are sparse, meaning that a vanishing fraction of possible edges is present, and many networks are arguably in the category that might be called "extremely sparse," where the number of edges increases only linearly with n. It is plausible, for instance, that the average number of friends a person has remains roughly constant even as the population of the world increases, so that the total number of edges in the network of friendships increases linearly with n.

One might hope that the result of Bollobás *et al.* would still remain true, at least approximately, for sparse networks, but a simple counterexample demonstrates that this is not the case. Consider the classic sparse model network known as the (Poisson) random graph [16], in which n nodes are connected by edges placed independently at random between every distinct node pair with probability $c/(n-1)$ for some constant c. It is straightforward to demonstrate that the threshold occupation probability for percolation on such a graph falls at $1/c$ in the limit of large n [12, 13, 17]. The leading eigenvalue of the adjacency matrix, on the other hand, is bounded below by the square root of the largest degree [18], which goes as $\sqrt{\log n/\log\log n}$ and hence diverges as $n\to\infty$, so that the dense-graph result of [15] is in error by an infinitely wide margin in the limit.

Clearly then, a different result must hold for sparse graphs. In this paper we give such a result. We show that the equivalent of the result of Bollobás *et al.* for sparse graphs is that the threshold occupation probability is equal to the inverse of the leading eigenvalue of the so-called Hashimoto or non-backtracking matrix [19, 20], an alternative matrix representation of network structure that has found recent use in studies of community detection and localization in networks [20, 21]. This result is exact for locally tree-like networks, which includes random graphs and many others, and applies not only for ensemble models of networks, such as the random graph, but also for individual networks, such as networks measured in empirical studies. For networks that are not locally tree-like the calculation does not give an exact figure for the percolation threshold, but we show that it provides a lower bound, which can still be of substantial use.

Our analysis is based on a formulation of the percolation problem as a message passing process. Message passing methods, also called cavity methods, are widely used in statistical physics and computer science for the solution of problems on networks [22, 23]. In the most common "belief propagation" formulation, the messages are single numbers representing probabilities, which are passed among the nodes of the network. In our approach the messages are instead generating functions of a single independent variable. The derivation is as follows.

Consider a bond percolation process on an arbitrary undirected network of n nodes and m edges. Edges are occupied uniformly at random with probability p or unoccupied with probability $1-p$. The primary entities of interest are the percolation clusters, sets of nodes connected by occupied edges. Since percolation is a random process, one cannot know with certainty the identity of

the clusters ahead of time, or even their size or number, but some things are known. In general there will (with high probability) be at most one percolating cluster, a cluster that fills a non-vanishing fraction of the network in the limit of large n, plus an extensive number of small clusters of finite average size. The percolating cluster appears only for sufficiently large values of p and the percolation threshold p_c is the value above which it appears; below p_c there are only small clusters.

Let us define $\pi_i(s)$ to be the probability that node i belongs to a small cluster of exactly s nodes, averaged over many realizations of the random percolation process. If the network is a perfect tree—if it contains no loops—then the size s of the cluster is equal to one (for node i itself) plus the sum of the numbers of nodes reachable along each edge attached to i, which is zero if the edge is unoccupied or nonzero otherwise. If, on the other hand, there are loops in the network then this calculation will not in general give the exact value of s, since it may be possible to reach the same node along two different occupied edges, which leads to overcounting. If the network satisfies the weaker condition of being *locally tree-like*, however, meaning that in the limit of large network size an arbitrarily large neighborhood around any node takes the form of a tree, then our calculation gives a good approximation, which becomes exact in the $n \to \infty$ limit.

Working in the large n limit then and assuming the network to be locally tree-like, the probability $\pi_i(s)$ is equal to the probability that the numbers of nodes reachable along each edge from i add up to $s-1$, which we can write as

$$\pi_i(s) = \prod_{j\in\mathcal{N}_i} \sum_{s_j=0}^{\infty} \pi_{i\leftarrow j}(s_j)\, \delta\biggl(s-1, \sum_{j\in\mathcal{N}_i} s_j\biggr), \tag{1}$$

where $\pi_{i\leftarrow j}(s)$ is the probability that exactly s nodes are reachable along the edge connecting i and j, $\mathcal{N}_i$ is the set of immediate network neighbors of node i, and $\delta(a,b)$ is the Kronecker delta. The delta function ensures that only those terms in which the number of reachable nodes add up to exactly $s-1$ are included in the sum.

We now introduce a probability generating function $G_i(z) = \sum_{s=1}^{\infty} \pi_i(s)\, z^s$, whose value is given by

$$\begin{aligned} G_i(z) &= \sum_{s=1}^{\infty} z^s \prod_{j\in\mathcal{N}_i} \sum_{s_j=0}^{\infty} \pi_{i\leftarrow j}(s_j)\, \delta\biggl(s-1, \sum_{j\in\mathcal{N}_i} s_j\biggr) \\ &= z \prod_{j\in\mathcal{N}_i} \sum_{s_j=0}^{\infty} \pi_{i\leftarrow j}(s_j)\, z^{s_j}, \end{aligned} \tag{2}$$

which can be conveniently written as

$$G_i(z) = z \prod_{j\in\mathcal{N}_i} H_{i\leftarrow j}(z), \tag{3}$$

where $H_{i\leftarrow j}(z) = \sum_{s=0}^{\infty} \pi_{i\leftarrow j}(s)\, z^s$ is the generating function for $\pi_{i\leftarrow j}(s)$.

To calculate the $\pi_{i\leftarrow j}(s)$, we note that $\pi_{i\leftarrow j}(s)$ is zero if the edge between i and j is unoccupied (which happens with probability $1-p$) and nonzero otherwise (probability p), which means that $\pi_{i\leftarrow j}(0) = 1-p$, and for $s \geq 1$

$$\pi_{i\leftarrow j}(s) = p \prod_{k\in\mathcal{N}_j\setminus i} \sum_{s_k=0}^{\infty} \pi_{i\leftarrow j}(s_k)\, \delta\biggl(s-1, \sum_{k\in\mathcal{N}_j\setminus i} s_k\biggr), \tag{4}$$

where the notation $\mathcal{N}_j\setminus i$ denotes the set of neighbors of j excluding i. Substituting this expression into the definition of $H_{i\leftarrow j}(z)$ above, we then find that

$$H_{i\leftarrow j}(z) = 1 - p + pz \prod_{k\in\mathcal{N}_j\setminus i} H_{j\leftarrow k}(z). \tag{5}$$

This self-consistent equation for the generating function $H_{i\leftarrow j}(z)$ suggests a message-passing algorithm for calculating the distribution of percolation cluster sizes: for any chosen value of z one guesses (for instance at random) an initial set of values for the $H_{i\leftarrow j}$ and feeds them into the right-hand side of Eq. (5), giving a new set of values on the left. Repeating this process to convergence gives a solution for the generating functions, which can then be substituted into Eq. (3) to give the generating function for the cluster probabilities $\pi_i(s)$, from which we can recover the probabilities themselves by differentiating.

As an example application of the method, note that, since $\pi_i(s)$ is the probability that i belongs to a small (non-percolating) cluster of size s, the probability that it belongs to a small cluster of any size is $\sum_s \pi_i(s) = G_i(1)$ and the probability that it belongs to the percolating cluster is one minus this. Then the expected fraction S of the network occupied by the entire percolating cluster is given by the average over all nodes:

$$S = \frac{1}{n}\sum_{i=1}^{n}\bigl[1 - G_i(1)\bigr] = 1 - \frac{1}{n}\sum_{i=1}^{n} \prod_{j\in\mathcal{N}_i} H_{i\leftarrow j}(1). \tag{6}$$

Setting $z=1$ in Eq. (5) we have

$$H_{i\leftarrow j}(1) = 1 - p + p \prod_{k\in\mathcal{N}_j\setminus i} H_{j\leftarrow k}(1), \tag{7}$$

and the solution of this equation, for instance by iteration from a random initial guess, allows us to calculate the size of the percolating cluster. Note that the number of quantities $H_{i\leftarrow j}(1)$ is $2m$—twice the number of edges m because $H_{i\leftarrow j}$ is distinct from $H_{j\leftarrow i}$—which is small enough to allow the numerical iteration of Eq. (7) quickly on the sparse networks that are our primary focus.

There do of course exist other algorithms for calculating the distribution of cluster sizes for percolation on networks, but these algorithms all calculate clusters for only a single realization of the randomness inherent in

the percolation process. The method described here returns the probability distribution of cluster sizes over all realizations of the randomness. To calculate this quantity by standard methods would require many runs of a traditional algorithm for different realizations, which could take a long time and would ultimately return only a stochastic approximation to the average and not an exact result.

In addition to providing an algorithm for calculating average percolation properties, however, our method also allows us to derive new fundamental results for network percolation by analyzing the expected behavior of the algorithm. In particular, we can calculate the exact position of the percolation threshold on an arbitrarily large, locally tree-like network, as follows.

The first derivative $H'_{i\leftarrow j}(z=1) = \sum_{s=0}^{\infty} s\pi_{i\leftarrow j}(s)$ is the expected number of (non-percolating-cluster) nodes reachable along the edge from i to j. Differentiating Eq. (5) and assuming we are below the percolation threshold $p < p_c$, we have

$$H'_{i\leftarrow j}(1) = p + p \sum_{k\in\mathcal{N}_j\setminus i} H'_{j\leftarrow k}(1), \tag{8}$$

where we have made use of the fact that $H_{i\leftarrow j}(1) = 1$ for all edges i, j below the threshold. Defining a $2m$-element vector $\mathbf{h}$ whose elements are equal to $H'_{i\leftarrow j}(1)$, Eq. (8) can be written in matrix form as $\mathbf{h} = \sqrt{2m}\,p\mathbf{1} + p\mathbf{B}\mathbf{h}$, or equivalently $\mathbf{h} = \sqrt{2m}\bigl(p^{-1}\mathbf{I} - \mathbf{B}\bigr)^{-1}\mathbf{1}$, where $\mathbf{1}$ is the properly normalized uniform vector $(1,1,1,\ldots)/\sqrt{2m}$, $\mathbf{I}$ is the identity, and $\mathbf{B}$ is a $2m \times 2m$ non-symmetric matrix with rows and columns indexed by directed edges $i \leftarrow j$ and elements $B_{i\leftarrow j,k\leftarrow l} = \delta_{jk}(1-\delta_{il})$. Then the average number of nodes reachable along any edge in the network is

$$\begin{aligned}\frac{1}{2m}\sum_i \sum_{j\in\mathcal{N}_i} H'_{i\leftarrow j}(1) &= \frac{1}{\sqrt{2m}}\mathbf{1}^T\mathbf{h} \\ &= \mathbf{1}^T\bigl(p^{-1}\mathbf{I} - \mathbf{B}\bigr)^{-1}\mathbf{1} = \sum_{\nu=1}^{2m} \frac{\mathbf{1}^T\mathbf{v}_\nu\mathbf{u}_\nu^T\mathbf{1}}{p^{-1}-\lambda_\nu},\end{aligned} \tag{9}$$

where λ_ν is the νth eigenvalue of $\mathbf{B}$ and $\mathbf{u}_\nu$ and $\mathbf{v}_\nu$ are the corresponding left and right eigenvectors. The matrix $\mathbf{B}$ is known as the non-backtracking matrix and has been a focus of recent attention for its role in community detection and centrality calculations on networks [20, 21].

As we approach the percolation threshold from below, the average number of reachable nodes grows because the small clusters are growing, and it diverges precisely at the percolation threshold where the average cluster size diverges. From Eq. (9) we see that this happens when p^{-1} equals the largest eigenvalue of the non-backtracking matrix and hence we conclude that *the critical percolation probability p_c of a sparse, locally tree-like network is equal to the reciprocal of the leading eigenvalue of the non-backtracking matrix.*

As a simple example consider a random k-regular graph, i.e., a network in which every node has exactly k edges but connections are otherwise made at random. For such a graph the non-backtracking matrix has $k-1$ nonzero elements in each row and column and hence its largest eigenvalue is exactly $k-1$, giving $p_c = 1/(k-1)$, which can easily be confirmed to be the correct answer using other methods [12, 13]. The leading eigenvalue of the adjacency matrix on the other hand, which gives the percolation threshold for dense graphs as described in the introduction, is k in this case, and hence would give a lower, and incorrect, result of $p_c = 1/k$.

In fact, the leading eigenvalue of the adjacency matrix is never less than the leading eigenvalue of the non-backtracking matrix. To see this, we define $F_j(z) = 1 - p + pz\prod_{k\in\mathcal{N}_j} F_k(z)$, which differs from the equation for our generating function $H_{i\leftarrow j}(z)$, Eq. (5), only in that node i is not omitted from the product. Differentiating as we did in Eq. (8) we see that

$$F'_j(1) = p + p\sum_{k\in\mathcal{N}_j} F'_k(1) = p + p\sum_k A_{jk}F'_k(1), \tag{10}$$

where A_{jk} is an element of the adjacency matrix. In matrix notation this can be written as $\mathbf{F'} = p\mathbf{1} + p\mathbf{A}\mathbf{F'}$ and hence $\mathbf{F'} = (p^{-1}\mathbf{I} - \mathbf{A})^{-1}\mathbf{1}$, which diverges when p equals the reciprocal of the leading eigenvalue of the adjacency matrix. On the other hand, if we solve Eq. (10) by iteration starting from an initial value of $F'_j(1) = \max_i H'_{i\leftarrow j}(1)$, it is straightforward to see that $F'_j(1)$ can never decrease, and hence $F'_j(1) \geq H'_{i\leftarrow j}(1)$ always, for all i, j. If we imagine therefore increasing the value of p slowly from zero towards the percolation threshold, $F'_j(1)$ must diverge at or before the point at which $H'_{i\leftarrow j}(1)$ diverges, and hence the leading eigenvalue of the adjacency matrix is greater than or equal to that of the non-backtracking matrix. This in turn implies that *the dense-matrix result for the percolation threshold based on the adjacency matrix is a lower bound on the percolation threshold of a sparse graph.*

An interesting special case is that of a perfect tree, a network with no loops at all. Percolation, in the sense of a percolating cluster that fills a nonzero fraction of the network in the large-n limit, never occurs on such a network—for all $p < 1$ the largest cluster occupies only a vanishing fraction of the network and our formalism gives this result correctly. The diagonal elements of powers of the non-backtracking matrix count numbers of closed non-backtracking walks on a graph [20, 24] (hence the name "non-backtracking matrix"), but a perfect tree has no such walks, so the trace of every power of the matrix is zero and hence so also are all the eigenvalues. Thus the reciprocal of the largest eigenvalue diverges and there is no percolation threshold. The leading eigenvalue of the adjacency matrix, on the other hand, is nonzero on a tree. On a k-regular tree, for instance, the leading eigenvalue of the adjacency matrix for large n is k again, implying

a percolation threshold of $1/k$. This is, indeed, a lower bound on the true percolation threshold, as it must be, but it is in error by a wide margin.

In practical situations one can calculate the leading eigenvalue of the non-backtracking matrix numerically and invert to determine the percolation threshold, but the $2m \times 2m$ matrix can be quite large, making the calculation cumbersome. It can be sped up by using the so-called Ihara (or Ihara-Bass) determinant formula [20]. Let $\mathbf{v}$ be a right eigenvector of $\mathbf{B}$ as previously, and define $x_i = \sum_{j\in\mathscr{N}_i} v_{i\leftarrow j}$. Then we can show that the n-element vector $\mathbf{x} = (x_1, x_2, \ldots)$ satisfies the quadratic eigenvector equation

$$(\lambda^2 - 1 - \lambda\mathbf{A} + \mathbf{D})\mathbf{x} = 0, \tag{11}$$

where λ is an eigenvalue of the non-backtracking matrix and $\mathbf{D}$ is the diagonal matrix with the degrees of the nodes on the diagonal. Defining $\mathbf{w} = \lambda\mathbf{x}$, we then have

$$\mathbf{A}\mathbf{w} + (\mathbf{I} - \mathbf{D})\mathbf{x} = \lambda\mathbf{w}, \tag{12}$$

which can be rewritten as $\mathbf{M}\mathbf{z} = \lambda\mathbf{z}$ where $\mathbf{z} = (\mathbf{w}|\mathbf{x})$ and $\mathbf{M}$ is the $2n \times 2n$ matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{I}-\mathbf{D} \\ \mathbf{I} & \mathbf{0} \end{pmatrix}. \tag{13}$$

For a sparse network this matrix is sparse with only $2m + 2n$ nonzero elements—far fewer than the non-backtracking matrix itself—which permits rapid numerical calculation of the leading eigenvalue.

All of our results so far have been for tree-like networks, but most real-world networks are not trees. We can nonetheless use the techniques developed here to say something about the non-tree-like case. On a tree the number of nodes reachable along the edge from i to j is one (for node i itself) plus the sum of the numbers $n_{j\leftarrow k}$ reachable along every other edge attached to j. On a non-tree, on the other hand, this sum overestimates the number of reachable nodes because some nodes are reachable along more than one edge from j. This means that the generating function $H_{i\leftarrow j}(z)$ for the true number of reachable nodes will be larger than the value given by a naive estimate calculated by a simple average over the randomness:

$$\begin{aligned} H_{i\leftarrow j}(z) &\geq 1 - p + pz\Bigl\langle z^{\sum_{k\in\mathscr{N}_j\setminus i} n_{j\leftarrow k}} \Bigr\rangle \\ &= 1 - p + pz\Bigl\langle \prod_{k\in\mathscr{N}_j\setminus i} z^{n_{j\leftarrow k}} \Bigr\rangle \geq 1 - p + pz \prod_{k\in\mathscr{N}_j\setminus i} \bigl\langle z^{n_{j\leftarrow k}} \bigr\rangle, \end{aligned} \tag{14}$$

where the second inequality follows by an application of the Chebyshev integral inequality [25]. But $\bigl\langle z^{n_{j\leftarrow k}} \bigr\rangle = H_{j\leftarrow k}(z)$ by definition, so we find that on a non-tree-like network the exact equality of Eq. (5) is replaced with an inequality:

$$H_{i\leftarrow j}(z) \geq 1 - p + pz \prod_{k\in\mathscr{N}_j\setminus i} H_{j\leftarrow k}(z). \tag{15}$$

Suppose, however, that we nonetheless decide to use the exact equality of (5), iterating to estimate the generating functions. If we start from an initial value of $H_{i\leftarrow j}$ equal to the true answer we are looking for (which we don't know, but let us suppose momentarily that we do), then it is straightforward to see from (15) that the value of $H_{i\leftarrow j}$ will never increase under the iteration, implying that the value we calculate will be a lower bound on the true value for all z. As we approach the percolation threshold from above in the large size limit, the true value of $H_{i\leftarrow j}(1)$, which represents the probability that the edge from i to j connects to a small cluster, approaches 1, while the value calculated from Eq. (5), which is less than or equal to the true value, must reach 1 later, i.e., at a lower or equal value of p but never higher. Thus the percolation threshold estimated from (5) is never higher than the true percolation threshold. Equivalently, we can say that *for any network, p_c is always greater than or equal to the inverse of the leading eigenvalue of the non-backtracking matrix.* Thus the leading eigenvalue gives us a bound on the percolation threshold.

We can also combine this result with our earlier observation that the leading eigenvalue of the adjacency matrix is never less than that of the non-backtracking matrix to make the further statement that *p_c for any network is always greater than or equal to the inverse of the leading eigenvalue of the adjacency matrix.* Thus, both eigenvalues place lower bounds on p_c, but the bound given by the non-backtracking matrix is better (or at least never worse) than the one given by the adjacency matrix.

Table I shows tests of these result on a range of sparse networks. For each network we have computed (an ap-

	Percolation threshold		
Network	Adjacency	Non-backtracking	Actual
Random graph	0.161	0.200	0.200
Block model	0.140	0.167	0.173
Circuit	0.200	0.340	0.47
Internet	0.0140	0.0155	0.0231
Gnutella	0.0759	0.0871	0.097
Amazon	0.0426	0.0562	0.10

TABLE I: Percolation thresholds estimated from the eigenvalues of the adjacency and non-backtracking matrices, and measured directly in numerical simulations (or calculated exactly in the case of the random graph). The networks are: a Poisson random graph with average degree 5 and 100 000 nodes; a stochastic block model with 100 000 nodes, four groups, and an average of 4 in-group and 2 out-group edges per node; electronic circuit 838 from the ISCAS 89 benchmark set [26]; a snapshot of the Internet autonomous system peering structure; a Gnutella peer-to-peer filesharing network [27]; and a copurchasing network of items on Amazon.com [28].

proximation to) the true percolation threshold by repeated numerical simulations and the bounds given by the leading eigenvalues of the non-backtracking and adjacency matrices. As the table shows, in each case the non-backtracking matrix does indeed give a lower bound on the true threshold, and in each case it gives a more accurate estimate than the adjacency matrix.

In summary, we have in this paper studied the problem of percolation on an arbitrary network and shown that for locally tree-like networks percolation can be reformulated as a message passing process, allowing us to solve for average percolation properties such as the size of the percolating cluster. In addition, by analyzing the message passing equations themselves, we have shown that the position of the percolation threshold is given by the inverse of the leading eigenvalue of the non-backtracking matrix. On non-tree-like networks the message passing approach is not exact, but it gives bounds on the exact results.

The authors thank Cris Moore for useful conversations. This work was funded in part by the National Science Foundation under grant DMS–1107796 and by the Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA) under grant FA9550–12–1–0432.

After this work was completed we learned of unpublished work by Hamilton and Pryadko [29] that independently demonstrates some of the same results.

[1] J. Machta, Phase transition in fractal porous media. *Phys. Rev. Lett.* **66**, 169–172 (1991).

[2] K. Moon and S. M. Girvin, Critical behavior of superfluid ^{4}He in aerogel. *Phys. Rev. Lett.* **75**, 1328–1331 (1995).

[3] T. Odagaki and S. Toyofuku, Properties of percolation clusters in a model granular system in two dimensions. *J. Phys. Cond. Mat.* **10**, 6447–6452 (1998).

[4] J. Tobochnik, Granular collapse as a percolation transition. *Phys. Rev. E* **60**, 7137–7142 (1999).

[5] S. de Bondt, L. Froyen, and A. Deruyttere, Electrical conductivity of composites: A percolation approach. *J. Mater. Sci.* **27**, 1983–1988 (1992).

[6] D. P. Bentz and E. J. Garboczi, Modelling of the leaching of calcium hydroxide from cement paste: Effects on pore space percolation and diffusivity. *Materials and Structures* **25**, 523–533 (1992).

[7] L. de Arcangelis, S. Redner, and A. Coniglio, Anomalous voltage distribution of random resistor networks and a new model for the backbone at the percolation threshold. *Phys. Rev. B* **31**, 4725–4728 (1985).

[8] C. L. Henley, Statics of a self-organized percolation model. *Phys. Rev. Lett.* **71**, 2741–2744 (1993).

[9] A. Bunde, S. Havlin, and M. Porto, Are branched polymers in the universality class of percolation? *Phys. Rev. Lett.* **74**, 2714–2716 (1995).

[10] P. Grassberger, On the critical behavior of the general epidemic process and dynamical percolation. *Math. Biosci.* **63**, 157–172 (1983).

[11] M. E. J. Newman, Spread of epidemic disease on networks. *Phys. Rev. E* **66**, 016128 (2002).

[12] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin, Resilience of the Internet to random breakdowns. *Phys. Rev. Lett.* **85**, 4626–4628 (2000).

[13] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Lett.* **85**, 5468–5471 (2000).

[14] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, Attack vulnerability of complex networks. *Phys. Rev. E* **65**, 056109 (2002).

[15] B. Bollobás, C. Borgs, J. Chayes, and O. Riordan, Percolation on dense graph sequences. *Annals of Probability* **38**, 150–183 (2010).

[16] P. Erdős and A. Rényi, On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* **5**, 17–61 (1960).

[17] S. Janson, On percolation in random graphs with given vertex degrees. *Electronic Journal of Probability* **14**, 5 (2009).

[18] M. Krivelevich and B. Sudakov, The largest eigenvalue of sparse random graphs. *Combinatoric, Probability and Computing* **12**, 61–72 (2003).

[19] K. Hashimoto, Zeta functions of finite graphs and representations of p-adic groups. *Adv. Stud. Pure Math.* **15**, 211–280 (1989).

[20] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci. USA* **110**, 20935–20940 (2013).

[21] T. Martin, X. Zhang, and M. E. J. Newman, Localization and centrality in networks. Preprint arXiv:1401.5093 (2014).

[22] J. S. Yedidia, W. T. Freeman, and Y. Weiss, Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel (eds.), *Exploring Artificial Intelligence in the New Millennium*, pp. 239–270, Morgan Kaufmann, San Francisco, CA (2003).

[23] M. Mézard and A. Montanari, *Information, Physics, and Computation*. Oxford University Press, Oxford (2009).

[24] N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin, Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics* **9**, 585–603 (2007).

[25] B. Karrer and M. E. J. Newman, A message passing approach for general epidemic models. *Phys. Rev. E* **82**, 016101 (2010).

[26] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, Superfamilies of evolved and designed networks. *Science* **303**, 1538–1542 (2004).

[27] M. Ripeanu, I. Foster, and A. Iamnitchi, Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing* **6**, 50–57 (2002).

[28] J. Leskovec, L. Adamic, and B. A. Huberman, The dynamics of viral marketing. *ACM Trans. Web* **1**, 5 (2007).

[29] K. E. Hamilton and L. P. Pryadko, Tight lower bound for percolation threshold on a quasi-regular graph. Preprint arXiv:1405.0050 (2014).