SCIENTIFIC REPORTS

OPEN

Efficient message passing for cascade size distributions

Rebekka Burkholz[1,2,3]

How big is the risk that a few initial failures of networked nodes amplify to large cascades that endanger the functioning of the system? Common answers refer to the average final cascade size. Two analytic approaches allow its computation: (a) (heterogeneous) mean field approximation and (b) belief propagation. The former applies to (infinitely) large locally tree-like networks, while the latter is exact on finite trees. Yet, cascade sizes can have broad and multi-modal distributions that are not well represented by their average. Full distribution information is essential to identify likely events and to estimate the tail risk, i.e. the probability of extreme events. We therefore present an efficient message passing algorithm that calculates the cascade size distribution in finite networks. It is exact on finite trees and for a large class of cascade processes. An approximate version applies to any network structure and performs well on locally tree-like networks, as we show with several examples.

Mean field theories are core to the analysis of stochastic processes on networks, as they make them analytically tractable and allow the estimate of average quantities of interest. Fundamental to this approach is the configuration model and its variants[1]. These create random network ensembles whose locally tree-like network structure is exploited to approximate the average neuronal activity in a brain[2,3], estimate the size of an epidemic outbreak[4], measure systemic risk[5], or analyze the formation of opinions[6]. Their analysis has deepened our understanding of cascade phenomena and provided insights into the average role of connectivity in the spreading of failures or activations[7–10]. In consequence, many of our insights rely on Local Tree Approximations (LTA) and, thus, the assumption that large systems can be approximated well by their infinitely large counterpart and that neighbors of the same node are independent.

Finite systems that are small enough so that finite size effects have to be considered are subject of study in many important applications. For a given and fixed network, belief propagation (BP), also termed cavity method in Physics, serves the computation of average node states and thus the average final cascade size. Furthermore, it can provide means to estimate the probability of extreme events in large systems[11]. As LTA, BP relies on independent neighbors and is thus exact on trees, while an iterative application (i.e., loopy belief propagation) approximates well average cascade results on locally tree-like networks[12].

Yet, finite networks, even when they are large, can behave quite different from the expected, in particular close to phase transitions. Even for large systems, the distribution of the final cascade size can be broad and of multi-modal shape. This has been shown for specific topologies, i.e., complete networks and stars[13]. Another example is the well known Curie-Weiss model[14], whose magnetization density distribution is bi-modal for low temperature. Also real world applications elucidate the need for distribution information in addition to averages[15].

Multi-modality has been frequently observed close to phase transitions based on Monte Carlo simulations and also used to characterize them. Yet, also for parameters far away from sudden regime shifts, we can observe broad multi-modal distributions. BP or LTA analysis cannot capture these and report a single, potentially improbable event: the average of the distribution.

As alternative, we propose a message passing algorithm that computes the full cascade size distribution. Message passing is a natural algorithmic choice that lends itself to efficient parallelization and makes use of the fact that cascades emerge from local interactions. Also LTA[16] and BP[12] can be formulated according to this principle, yet, are structurally quite different. In contrast to BP, we only have to go through a tree once instead of twice. As in each node cascade size distributions of subtrees rooted in its children are combined, we term this approach *Subtree Distribution Propagation* (SDP). It is exact on trees and efficient. For limited resolution of the cascade size, it only requires a number of operations that is linear in the number of network nodes: $O(N)$. The exact approach scales as $O(N^2 log N)$. Yet, the involved convolutions are still approximated with the help of Fast Fourier Transformations. To further approximate the cascade size distribution on general networks, we introduce

[1]ETH Zurich, Chair of Systems Design, Zurich, 8092, Switzerland. [2]ETH Zurich, Computer Science, Zurich, 8092, Switzerland. [3]Harvard T.H. Chan School of Public Health, Biostatistics, Boston, MA, 02115, USA. Correspondence and requests for materials should be addressed to R.B. (email: rburkholz@hsph.harvard.edu)

a second algorithm: termed *Tree Distribution Approximation* (*TDA*). It relies on loopy belief propagation (or another algorithm to compute marginal activation probabilities of nodes) and SDP. By comparison with extensive Monte Carlo simulations, we show that TDA approximates the cascade size distribution on locally tree-like networks well. As we discuss further, our derivations can form the basis of algorithms for general network topologies.

Cascade Model Framework

We assume that a fixed undirected network (or graph) $G = (V, E)$ with node (or vertex) set V and link (or edge) set E is given. Each $i \in V$ of the $N = |V|$ nodes is equipped with a binary state $s_i \in \{0, 1\}$, where $s_i = 1$ indicates that i is active (or failed) and $s_i = 0$ that i is inactive (or functional). In the course of a cascade, node states can become activated by local interactions with network neighbors, i.e., the nodes a node is connected with by links. Note that activation can travel in both directions of a link. We assume that the process evolves over discrete time steps $t = 0, \cdots, T$ and that the activation of a node i at time t depends on the number $a_i(t-1)$ of active neighbors at the previous time step.

The respective cascade model is defined by the response functions R_i for each node $i \in V$. A node i becomes active with probability $R_i(a)$ when exactly a of its neighbors are active (while $a-1$ would not have been enough). Thus, i activates with probability $R_i(0)$ and never activates with probability $R_i(d_i+1)$, where d_i denotes i's degree, i.e., the number of its neighbors. We further define $R_i^c(a)$ as probability that a node becomes active whenever a neighbors are active. Usually, this is the cumulative sum $R_i^c(a) = \sum_{l=0}^{a} R_i(l)$ and we have $\sum_{a=0}^{d_i+1} R_i(a) = 1$. This reflects the reasoning that each active neighbor increases the chance to activate the node. For instance in opinion formation models, also opposite effects could be thought of, i.e., a high number of active neighbors reduces the probability of adopting the same opinion. For simplicity, we assume that R_i is not time dependent itself and exclude the possibility of recovery, i.e., that a node switches from an active/failed ($s_i = 1$) back to an inactive/functional state ($s_i = 0$). In principle, the recovery of a node could be considered by the introduction of a third node state $s_i = $ 'recovered', but would introduce additional computational complexity that we avoid here.

In this setting, we are interested in the final cascade size that is measured by the final fraction of active nodes $\rho = \frac{1}{N}\sum_{i=1}^{N} s_i(T)$. It answers, for instance, the question how many nodes receive a certain information or how many pass on a disease. Regardless whether we want to minimize or maximize ρ, considering the probability of adverse events can improve the decision making.

This framework covers many cascade models, ranging from neural dynamics to Voter models[17,18]. Two common examples shall be discussed in more detail: (a) a threshold model (TM) of information propagation[6,19]. and (b) a simple model of epidemic spreading, also termed independent cascade model (ICM)[4,20,21]. Details are provided in the method section.

Message Passing for Cascade Size Distributions

In the following, we only provide an intuition for the main guiding principles of the two algorithms that we propose: subtree distribution propagation (SDP) and tree distribution approximation (TDA). SDP applies to trees only and is exact, while TDA deploys SDP to approximate the final cascade size distribution on a general network. Details and the main theorem that SDP works correctly are given in the method section.

Subtree distribution propagation (SDP). The goal is to compute the final cascade size distribution on a tree. To achieve this, the high level idea of SDP is to solve similar smaller problems repeatedly by computing subtree cascade size distributions and successively merging them. Hence, we chose the name subtree distribution propagation. In each node n, we calculate the cascade size distribution for the subtree T_n rooted in n dependent on the state of its parent p, and send this as message to p. Figure 1(a) visualizes the general procedure. We start in the leaves (i.e., the nodes with degree 1) at the highest level (i.e., at the bottom of the picture) and proceed iteratively upwards to the root r by combining the subtree distributions T_{c_i} corresponding to the children c_i.

In slight abuse of notation, let T_n denote the number of active nodes in the subtree rooted in n, which we also call subtree cascade size (as T_n/N). This is a random variable that can be expressed as sum over the node state s_n and children subtrees: $T_n = s_n + \sum_{i=1}^{d_n-1} T_{c_i}$. T_n and all involved node states depend on s_p (and each other) in complicated ways.

We control for this dependency by introducing an order-conditioning operator $||$ that has a similar function as conditioning on random variables. Yet, exact conditioning $T_n | s_p = 1$ would consider events where n causes the activation of p and vice versa. In contrast, we have to keep track of the right order of activations and do so with the help of $||$. $T_n || s_p$ removes the influence of n on p. We only focus on the impact of p on n. Specifically, $T_n || s_p$ denotes the cascade size of a tree T_n where the rest of the original network has been removed and n has an additional neighbor p, whose state is set to s_p with probability 1.

Computing the distribution of $T_n || s_p$ is challenging for two reasons: (a) the random variables are dependent and (b) the right order of activations needs to be respected. The solution for (a) is to order-condition T_n on events involving s_n (and s_p) that make the subtree distributions independent so that T_n is given by their convolution. Convolutions can be computed efficiently with the help of Fast Fourier Transformations (FFTs).

To solve (b), we define artificial variables I_n, A_n that capture the right order of activations and the dependence structure of s_n on s_p and T_{c_i}. I_n refers to an inactive and A_n to an active parent p. Key to the definition of A_n is an artificial node state visualized in Fig. 2 that considers whether n triggers the activation of its parent p. The distributions p_{I_n}, p_{A_n} are advanced iteratively so that we can assume their knowledge for the children I_{c_i}, A_{c_i}. Combined, they add the subtree cascade sizes and, separately, the number of active children a_n that can trigger the activation of n. Thus, in our subtree distribution propagation algorithm, each node (except the root) sends exactly one message to its parent: the distribution of I_n and A_n. This message is a combination and update of the messages the node received by its children, which is detailed in the method section. The root finally combines all received messages to compute the final cascade size distribution $p_\rho(x) = \mathbb{P}(T_r = xN)$.
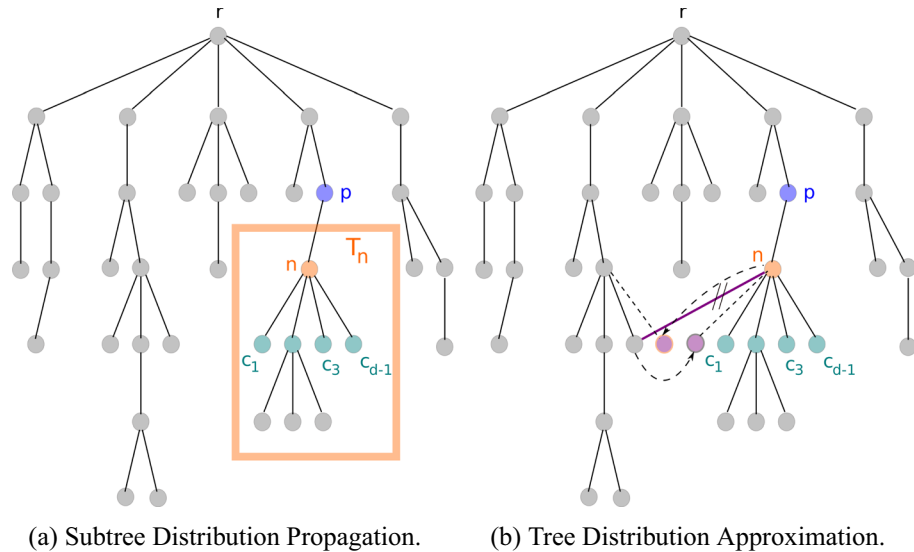
(a) Subtree Distribution Propagation. (b) Tree Distribution Approximation.

Figure 1. Illustration of relevant variables in the message passing algorithm. n denotes a focal node, p its parent, and T_n the subtree rooted in n. The calculation starts in the leaves (the bottom nodes with degree 1) and successively computes the cascade size distribution of each subtree T_n given the state of the parent p by combining the distributions corresponding to trees rooted in the children c_i. The resulting distribution is exact for a tree (**a**). If the network contains loops (**b**), (purple) links are deleted until a tree is obtained. Each deleted link is replaced by two new links that reconnect a independent (purple) copy of a cut-off neighbor. Such a copy is not counted as additional node in the final cascade size, but influences the activation probability of its neighbor n.
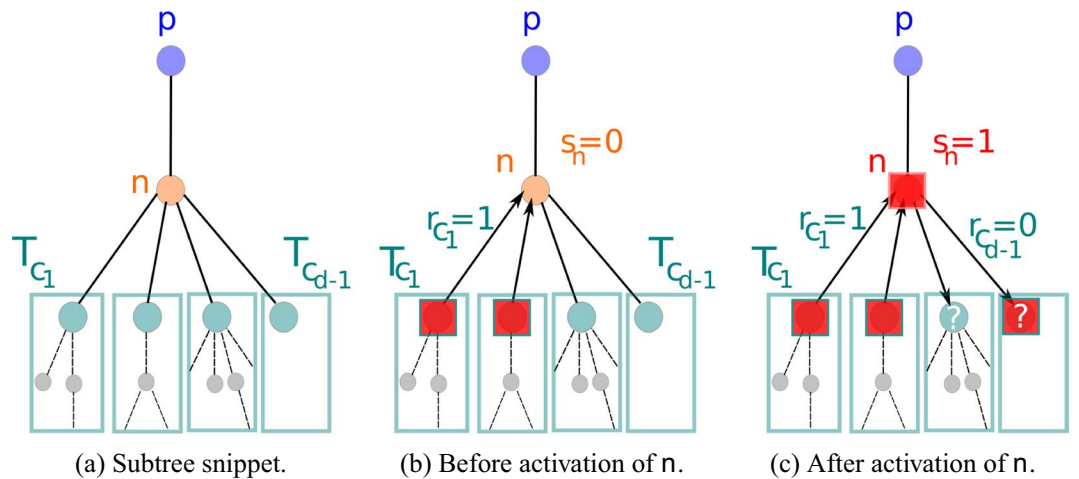

(a) Subtree snippet. (b) Before activation of n. (c) After activation of n.

Figure 2. Illustration of artificial state r_{c_i}. n denotes the focal node with degree d, p its parent, and $c_1, \ldots, c_{d-1}$ its children. T_{c_i} is a subtree rooted in a child c_i. An active node is represented by a red square. r_{c_i} denotes an artificial state of child c_i that indicates with $r_{c_i} = 1$ whether (**b**) it became active before its parent n and can thus trigger its activation or with $r_{c_i} = 0$ whether (**c**) it activates after its parent n or not at all.

This can be achieved computationally efficiently. The worst case algorithmic complexity of SDP is $O(N^2 \log N)$. Yet, for limited resolution of the cascade size. i.e., when we restrict ρ on an equidistant grid of $[0, 1]$, the algorithmic complexity of SDP is linear in the number of nodes: $O(N)$. It can further be brought down to $O(h)$, where h denotes the height of the tree, if the computations are distributed to computing units corresponding to nodes of the tree. A detailed analysis is provided in the Supplementary Information.

Tree distribution approximation (TDA). SDP is exact on trees. However, activations are stronger coupled in the presence of loops and the probability of large and small cascades tends to increase[13] so that the variance of the cascade size distribution grows. To take this into account, we propose an approximation version of SDP.

The idea is to first calculate individual activation probabilities on the original network and second to use them for adapting the response functions R_i. These are given as input to SDP which is applied to a minimum spanning

tree of the original network. Since this approach is only approximate and is based on the cascade size distribution on a tree, we call it tree distribution approximation (TDA).

In detail, we employ loopy BP to calculate the activation probabilities $p_{in} = \mathbb{P}(s_i = 1 || s_n = 0)$ of a neighbor i given that n is not active (before) to update the response function R_n, as outlined in the method section. Loopy BP itself is not exact, yet, usually approximates p_{in} well on locally tree-like networks. It could be substituted by any alternative algorithm. For instance, the Junction Tree Algorithm[22] would be exact (for directed acyclic graphs) but computationally costly and does not scale to large networks.

Next, we compute a minimum spanning tree of the original network (i.e., delete links of loops until we obtain a tree). Further, we assume that lost neighbors i of a node n activate initially and independently (before n) with probability p_{in} so that they can still contribute to the activation of n. Figure 1(b) illustrates this approach. We create an independent copy of a lost neighbor i (which is colored purple) and connect it with n. The copy's activation is not counted in the final cascade size ρ. It only influences the response R_n.

Therefore, this algorithm neglects certain dependencies of node activations in the presence of loops. If these loops are large enough, their contribution is usually negligible. We expect to approximate cascade size distributions well on locally tree-like networks. Next, we test this claim in numerical experiments.

Numerical Experiments

We focus on three exemplary networks that are representative of different use cases and visualized in Fig. 3: a tree, a locally tree-like network constructed by a configuration model with power law degree distribution, and a real world network defined by data on corporate ownership relationships[23], which is is locally tree-like. For each network, we compare the cascade size distributions obtained by our message passing algorithm, i.e., SDP for the tree and TDA for the two other networks, with Monte Carlo simulations.

We present results for the two introduced cascade models with the same parameter setting for all networks as specified in the method section. This provides a proof of concept and allows us to assess the approximation quality of TDA in Fig. 3. Results for further parameter settings and larger configuration model graphs are reported in the Supplementary Information.

First, we observe that SDP and TDA match perfectly the cascade size distributions obtained by extensive Monte Carlo simulations for the tree and the locally tree-like corporate ownership network. For the power law configuration model, where the task is much harder, TDA identifies the modes correctly, yet, tends to slightly underestimate the variance of the cascade size distribution. A considerable number of loops introduces additional correlations of note states that we cannot capture by our tree approximation.

Second, we note the broad cascade size distributions. This is unexpected by heterogeneous mean field or BP analysis, as our parameter choices for the cascade models are in no case critical: Neither does the average cascade size undergo a phase transition close to the chosen parameters in an infinitely large network with the same degree distribution as the original network, nor does the average cascade size change abruptly in the finite network for small changes in the parameters.

For the threshold model, we also observe several modes of the distribution on the tree and corporate ownership network. Clearly, the average cascade size does not represent the cascade risk well in these cases. Our approaches, SDP and TDA, add cascade size distribution information. These are useful in particular when we face star structures or, similarly, pronounced hubs (i.e., nodes with large degree), as these contribute to multiple distribution modes. The modes roughly correspond to events where no hub activates, one hub activates (so that many of its neighbors follow), two activate, etc., while longer paths have a smoothing effect on the distribution.

The independent cascade model shows single modes only, since we analyze parameters here where it is very likely that the center becomes active but does not substantially increase the activation probability of its neighbors. A priori, the precise shape of the cascade size distribution for complicated network structures is not clear and calls for a detailed analysis with the provided tools.

Discussion

We have introduced two algorithms that compute the final cascade size distribution for a large class of cascade models: (a) the subtree distribution propagation (SDP) is exact on trees, while (b) the tree distribution approximation (TDA) provides an approximation variant that applies to every network, yet, performs well for locally tree-like network structures.

Their derivation is based on two basic ingredients: artificial random variables that consider the right order of activations and an order-conditioning operation where the network above a node's parents are cut off. The latter creates an independence of subtree cascade size distributions, which enables their efficient combination. For limited resolution of the cascade size distribution, thus an approximate version of the algorithm, the SDP part of the algorithms is linear in the number of nodes $O(N)$ and can be distributed along the tree structure of the input. Each node needs to be visited only once. In consequence, the introduced algorithms are quite efficient and scalable.

As we argue, cascade size distribution information is critical for good decision making, when the distributions are broad and, in particular, when they have multiple modes, which signify probable events. Therefore, there is a need to generalize our approach beyond locally tree-like network structures, i.e., to networks with higher loop density. This generality will trade off with efficiency and scalability, similarly as the junction tree algorithm relates to belief propagation. The approach presented here lends itself as well for a transfer to junction trees.

On a meta level, we have presented a way to combine cascade size distributions of subnetworks and do not rely on the assumption that these subnetworks are trees themselves. Their distribution can either be computed analytically or approximated by Monte Carlo simulations. In every case, we can efficiently combine the related distributions if the subnetworks are connected in a tree-like fashion (as in junction trees).
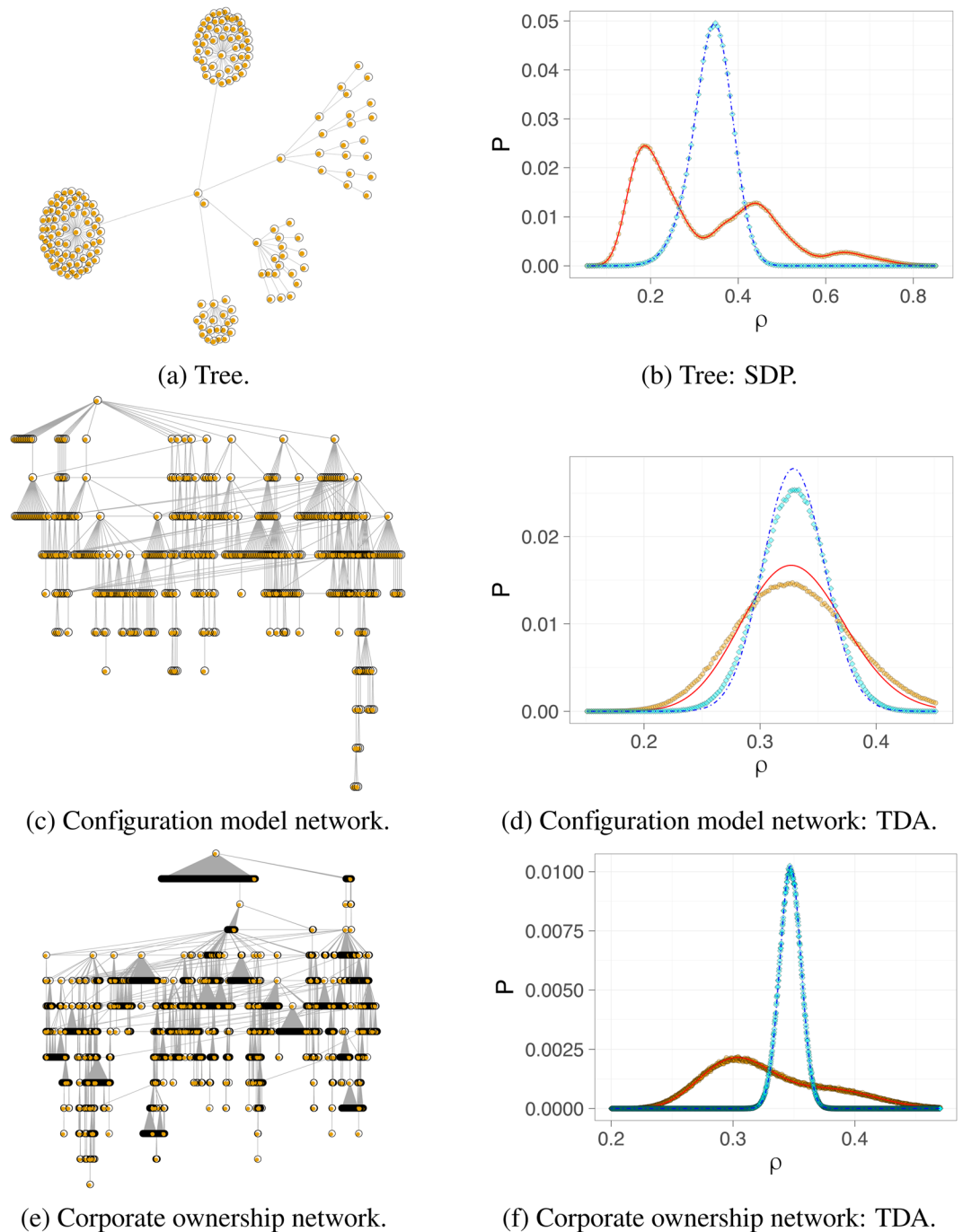
(a) Tree.

(b) Tree: SDP.

(c) Configuration model network.

(d) Configuration model network: TDA.

(e) Corporate ownership network.

(f) Corporate ownership network: TDA.

Figure 3. Cascade size distribution on exemplary networks. The left column shows the network, the right column the corresponding cascade size distributions. Symbols represent Monte Carlo simulations (with 10^6 realizations): orange circles for the threshold model and cyan squares for the independent cascade model. Lines correspond to the respective message passing algorithm: a solid red line for the threshold model and a blue dotted line for the independent cascade model.

Furthermore, the principle of our approach can be transferred to more general graphical models to obtain macro level information as, for instance, the distribution of the sum of involved random variables.

Materials and Methods
Cascade models.   We analyze two models in more detail, termed threshold model (TM) and independent cascade model (ICM). Both models have been used to describe similar phenomena, as information propagation, opinion formation, social influence, but also financial contagion or the spread of epidemics. While the cascade mechanisms are similar for both, an important distinction is that in the threshold model the probability to activate a neighbor depends on the other activations of neighbors[18].

Threshold model. The threshold model originates in a model of collective action[19], which has been transferred to networks by[6]. Each node i is equipped with a threshold θ_i that has been drawn initially independently at random from a distribution with cumulative distribution function F_i. Nodes with a negative threshold become active initially. Otherwise, a node activates whenever the fraction of active neighbors exceeds its threshold, i.e., $\theta_i \leq a/d_i$. In consequence, previous activations of neighbors influence the probability whether a further activation of a neighbor causes the activation of the focal node i. This implies a response function of the form:

$$R_i(0) = F_i(0), \quad R_i(a) = F_i\left(\frac{a}{d_i}\right) - F_i\left(\frac{a-1}{d_i}\right), \quad R_i(d_i + 1) = 1 - F_i(1), \quad R_i^c(a) = F_i\left(\frac{a}{d_i}\right).$$

Independent cascade model. The independent cascade model can be interpreted as simple epidemic spreading model that resembles the widely studied SIR (Susceptible-Infected-Recovered) model[20]. It is also equivalent to bond percolation in terms of the final outcome[4]. The activation of a node corresponds to its infection. Even though we do not explicitly allow for node recovery, for large networks, it can be implicitly incorporated in the choice of the infection probability p, i.e., the probability that a newly infected (active) node spreads a disease to a network neighbor. All neighbors of a newly infected node are infected independently. Also initially, nodes are activated independently with probability p. Thus, a node with degree d_i has the response function:

$$R_i(a) = p(1-p)^a, \quad R_i(d_i+1) = (1-p)^{d_i+1}, \quad R_i^c(a) = 1 - (1-p)^{a+1}$$

for $0 < a \leq d_i$. A node becomes activated exactly with a active neighbors (if it is not active initially, which is the case with probability $1-p$) and one out of the a neighbors causes the activation with probability p, while the remaining $a-1$ did not cause the activation.

Average cascade properties have been extensively studied for both models with the help of heterogeneous mean field approximations (for TM[7–10,24], for ICM[4]) and belief propagation (for TM[16], for more complicated variants than ICM[25,26]).

Numerical experiments. We run experiments for three networks that are visualized in Fig. 3. A tree and the configuration model network are created artificially, while the last one is a real world example based on data.

The tree consists of $N = 181$ nodes with two main hubs of degrees 69 and 50, while the configuration model network is a bit larger with $N = 543$ nodes and average degree $d_{avg} = 2.25$, but smaller maximal degree $d_{max} = 25$. The latter has degree distribution $p(d) \propto d^{-2.5}$, which is structurally similar to many real world networks[27,28]. Log-normally distributed degrees would have been sensible choice[28] as well, yet, the difference in degree distributions does not affect the cascade size distribution information critically. While the configuration model constructs locally tree-like networks, the size $N = 543$ is chosen on purpose relatively small so that the network has still a number of short loops as visible in Fig. 3(c). This makes our approximation task harder and serves as stress test for our approach.

The largest considered network is the largest weakly connected component of a publicly available network, which is defined by corporate ownership relationships[23]. It consists of $|V| = 4475$ nodes with mean degree $z = 2.08$ and maximal degree $d_{max} = 552$ and is clearly locally tree-like.

We compare the final cascade size distribution given by our algorithms with the results of Monte Carlo simulations always for the two introduced cascade models with the same parameter setting. In the threshold model, we assume independently normally distributed thresholds with a given mean $\mu = 0.5$ and standard deviation $\sigma = 0.5$ so that $F_i(\theta) = \Phi((x-\mu)/\sigma)$ for all $i \in V$, where Φ denotes the standard normal cumulative distribution function. The parameter p in the independent cascade model is always set to $p = 0.2$. This parameter choice is non-critical and thus, no phase transitions occur in close neighborhood of the parameters.

For Monte Carlo simulations, we always report the empirical distribution of 10^6 independent realizations. We calculate the final cascade size distribution for the tree by SDP and for the other two locally tree-like networks by TDA at full resolution, i.e., $\rho \in \{0, 1/N, 2/N, \ldots, 1\}$.

Subtree distribution propagation. The goal is to compute the final cascade size distribution $p_\rho(t/N) = \mathbb{P}(T_r = t)$ for a given tree $G = T_r$ with root r and cascade model with response functions R_i by a message passing algorithm. As explained in the main text, nodes n send messages p_{I_n}, p_{A_n} to their parent p, where I_n refers to an inactive ($s_p = 0$) and A_n to an active parent ($s_p = 1$). We define

$$I_n = (T_n, s_n)||s_p = 0,$$
$$A_n = (\widetilde{T}_n, r_n) = (r_n T_n \, _{\{s_n=1\}}||s_p = 0 + (1 - r_n)T_n \, _{\{s_p=1 \to s_n=1 \vee s_n=0\}}||s_p = 1, r_n),$$

where $r_n \in \{0, 1\}$ denotes an artificial state that indicates whether n triggers the activation of the parent p. Note that these are technically not random variables, as their distributions are not normalized. How their distributions are computed is stated later in Theorem 1. First, we explain their definition intuitively. To understand why I_n and A_n carry the right information, we shift the focus from n to its children and show how the messages corresponding to them enable us to compute the distribution of $T_n = s_n + \sum_{i=1}^{d_n-1} T_{c_i}$.

Let's first discuss the easier case when n stays inactive ($s_n = 0$). The subtree distributions of $T_{c_i}||s_n = 0$ are independent. Thus, we can convolute the distributions of $I_{c_i} = (T_{c_i}, s_{c_i})||s_n = 0$ to obtain the distribution of

$(T_n, a_n)||s_n = 0$ with $a_n = \sum_{i=1}^{d_n-1} s_{c_i}$. In this case, we know the probability that s_n does not become active (given its parent p): $\mathbb{P}(s_n = 0||a_n, s_p) = 1 - R_n^c(a_n + s_p)$.

The case $s_n = 1$ is more involved, since we have to consider only the children that trigger the activation of n, i.e., that become active before n. We therefore introduce an artificial binary node state r_n, which is illustrated by Fig. 2. $r_n = 1$ indicates that node n activates before its parent p and contributes to its activation, while $r_n = 0$ subsumes all other cases leading to $s_p = 1$, i.e., n does not activate before its parent, has an active parent, and might become active or not after the activation of its parent. We join r_n with an adapted subtree cascade size $\widetilde{T}_n$ to $A_n = (\widetilde{T}_n, r_n)$ so that $\sum_{i=1}^{d_n-1} A_{c_i} = (T_n - 1, a_n)||s_n = 1$ with now $a_n = \sum_{i=1}^{d_n-1} r_{c_i}$. $\widetilde{T}_n$ depends on r_n and s_n and is defined as $\widetilde{T}_n = r_n T_n 1_{\{s_n=1\}}||s_p = 0 + (1 - r_n)T_n 1_{\{s_p=1 \to s_n=1 \vee s_n=0\}}||s_p = 1$. Thus, if $r_n = 1$, n is active ($s_n = 1$) and T_n is not influenced by its parent, i.e., $s_p = 0$ is given. If $r_n = 0$, the parent is assumed to be active $s_p = 1$ and the node itself can either be inactive $s_n = 0$ or, if it activates ($s_n = 1$), p contributes to its activation so that n did not become active before p. Technically, A_n is not a random variable, since it is not normalized. Yet, its convolution still counts the right cases, which are input to the subtree cascade size distribution for active node n given its parent: $\mathbb{P}(T_n, s_n = 1||s_p)$.

In summary, the SDP starts in the bottom of a tree and computes messages p_{I_n}, p_{A_n} in each node, sends them to the parent p until the final cascade size distribution can be computed in the root. We make this reasoning explicit with the following theorem.

Theorem 1 *Let $G = (V, E)$ be a tree and R_i, R_i^c for $i \in V$ response functions defining a cascade model. The final cascade size distribution $p_p(t/N) = \mathbb{P}(T_r = t)$ is given by the result of a message passing algorithm ending in the root r, where at each node $n \in V$, the following computations are performed based on $p_{A_{c_i}}$, $p_{I_{c_i}}$ received from their children:*

Case $d_n = 1$ (leaves):

$$
\begin{aligned}
p_{A_n}(0, 0) &= \mathbb{P}(T_n = 0||s_p = 1) = \mathbb{P}(s_n = 0||s_p = 1) = 1 - R_n^c(1) \\
p_{A_n}(1, 0) &= \mathbb{P}(T_n = 1||s_p = 1) = R_n(1) \\
p_{A_n}(0, 1) &= \mathbb{P}(T_n = 0, s_n = 1||s_p = 0) = 0 \\
p_{A_n}(1, 1) &= \mathbb{P}(T_n = 1, s_n = 1||s_p = 0) = R_n(0) \\
p_{I_n}(1, 0) &= p_{I_n}(0, 1) = 0 \\
p_{I_n}(0, 0) &= \mathbb{P}(T_n = 0, s_n = 0||s_p = 0) = 1 - R_n(0) \\
p_{I_n}(1, 1) &= \mathbb{P}(T_n = 1, s_n = 1||s_p = 0) = R_n(0).
\end{aligned}
\tag{1}
$$

A node with degree $d_n > 1$ receives as input the distributions $p_{A_{c_i}}$, $p_{I_{c_i}}$ corresponding to its children. We define $p_{A_{n*}}$ and $p_{I_{n*}}$ as their 2-dimensional convolutions:

$$
\begin{aligned}
p_{A_{n*}}(t, f) &:= p_{A_{c_1}} ** p_{A_{c_2}} ** \cdots ** p_{A_{c_{d_n-1}}}[t, f] \\
p_{I_{n*}}(t, f) &:= p_{I_{c_1}} ** p_{I_{c_2}} ** \cdots ** p_{I_{c_{d_n-1}}}[t, f].
\end{aligned}
$$

Note that we have $p_{A_{n*}}(t, a) = p_{I_{n*}}(t, a) = 0$ for $t < a$.

Case $d_n > 1$, $n \neq r$:

$$
\begin{aligned}
p_{A_n}(t, 0) &= \mathbb{P}(T_n = t, s_n = 0||s_p = 1) \\
&\quad + \mathbb{P}(T_n = t; s_n = 1; s_p = 1 \to s_n = 1||s_p = 1) \\
&= \sum_{a=0}^{d_n-1} p_{I_{n*}}(t, a)(1 - R_n^c(a + 1)) + \sum_{a=0}^{d_n-1} p_{A_{n*}}(t - 1, a)R_n(a + 1) \\
p_{A_n}(t, 1) &= \mathbb{P}(T_n = t, s_n = 1||s_p = 0) = \sum_{a=0}^{d_n-1} p_{A_{n*}}(t - 1, a)R_n^c(a) \\
p_{I_n}(t, 0) &= \mathbb{P}(T_n = t, s_n = 0||s_p = 0) = \sum_{a=0}^{d_n-1} p_{I_{n*}}(t, a)(1 - R_n^c(a)) \\
p_{I_n}(t, 1) &= p_{A_n}(t, 1),
\end{aligned}
\tag{2}
$$

At root r:

$$
\begin{aligned}
\mathbb{P}(T_r = t) &= \mathbb{P}(T_r = t, s_r = 0) + \mathbb{P}(T_r = t, s_r = 1) \\
&= \sum_{a=0}^{d_r} p_{I_{r*}}(t, a)(1 - R_r^c(a)) + \sum_{a=0}^{d_r} p_{A_{r*}}(t - 1, a) R_r^c(a)
\end{aligned}
\tag{3}
$$

The proof of this theorem is given in the Supplementary Information along with a pseudocode of SDP.

Algorithmic complexity of SDP. A detailed discussion of the algorithmic complexity of SDP is provided in the Supplementary Information. In summary, computing the messages p_{I_n} and p_{A_n} in a node n requires $O(d_n(|T_n| + |T_n|\log(|T_n|)))$ computations, where $|T_n|$ denotes the number of nodes in the subtree rooted in n. Thus, in total $O(\sum_{n=1}^{N} d_n(|T_n| + |T_n| \log(|T_n|)))$ computations are needed to obtain the final cascade size distribution.

We have two options to reduce the run time: (a) limit the accuracy of the cascade size distribution so that $|T_n|$ can be substituted by a constant C. For instance, p_ρ can be defined only on an equidistant grid of $[0, 1]$. In this case, we are left with $O(\sum_{n=1}^{N} d_n) = O(N)$ computations.

(b) We can parallelize the matrix times vector multiplications, the Fast Fourier Transformations, and distribute the computations of messages for distinct nodes that are in different subtrees. A combination of (a) and (b) usually leads to an algorithm with smaller run time than $O(N)$, i.e., $O(h)$, where h refers to the height of a tree. In the worst case (for instance a long line), this can still require $O(N)$ computations.

Note that the choice of root is relevant for the run time of the algorithm. Minimizing the maximum path length from the root to any other node in the tree is beneficial in case that enough computing units are available for distribution of the work load. In addition, it can be advantageous to place nodes with high degree close to the root so that subtrees are kept small in the beginning. Convolutions related to those subtrees operate on small cascade sizes and thus require less computational effort.

Tree Distribution approximation. TDA employs SDP to approximate the final cascade size distribution on a general network $G = (V, E)$. First, we compute a minimum spanning tree M of G and run *SDP* on M with updated response functions $\widetilde{R}_i$. The algorithms consists of four main steps that are detailed next.

(1) We first compute the activation probability p_i of each node $i \in V$ by belief propagation on G. We therefore need to know how many neighbors activate before the node i. Each neighbor j activates with probability $p_{ij} = \mathbb{P}(s_i = 1||s_j = 0)$ before i and, according to our BP assumption, all neighbors activate independently. They fulfill the self-consistent equations:

$$
p_{ij} = \mathbb{P}(s_i = 1||s_j = 0) = \sum_{\mathbf{s}_{nb(i)\backslash j} \in \{0,1\}^{d_i-1}} R_i^c \left(\sum_{n \in nb(i)\backslash j} s_n \right) \prod_{n \in nb(i)\backslash j} p_{ni}^{s_n}(1 - p_{ni})^{1-s_n},
$$

where nb(i) denotes the set of neighbors of i and $\mathbf{s}_{nb(i)\backslash j}$ a vector consisting of states s_n of i's neighbors n except j. If G is a tree, the independence assumption is correct and we only need to visit each node twice to calculate the correct probabilities p_{ij}. Starting in the bottom of a tree, for each node n, we can compute p_{np} based on n's children, while its parent p has no influence on n. Next, we start in the root of the tree and proceed to compute p_{pn} until we reach the bottom. However, this is not enough if G is not a tree. Then, loopy BP interprets the equation above as system of fixed point equations (for p_{ij}) that we solve iteratively. A reasonable initialization is $p_{ij} = R_i(0)$.

For TDA, we always iterate 50 times through the whole network, which is enough to reach convergence in our cases. The product over neighbors is computed efficiently with the help of Fast Fourier Transformations. Based on p_{ij}, the activation probability of a node reads as

$$
p_i = \mathbb{P}(s_i = 1) = \sum_{\mathbf{s}_{nb(i)} \in \{0,1\}^{d_i}} R_i^c \left(\sum_{n \in nb(i)} s_n \right) \prod_{n \in nb(i)} p_{ni}^{s_n}(1 - p_{ni})^{1-s_n}.
$$

(2) We compute a minimum spanning tree $M = (V_M, E_M)$ of the original network G. We report results for a randomly chosen minimum spanning tree. However, weighting edges can give preference to which edges should be removed or kept, for instance, edges connecting nodes with larger degrees etc. Let us denote by $dnb(i) = \{j \in V | (i, j) \in E, (i, j) \notin E_M\}$ the set of neighbors of a node i in G that i is not connected to anymore in M, and let $m_i = |dnb(i)|$ be the number of such lost neighbors.

(3) Then, we update the response functions R_i of each node i by the probability that i activates after a of its neighbors in M activated. In addition, we assume that each of i's deleted neighbors n has activated initially with probability p_{ni}. We therefore consider the activation of the deleted neighbors as independent of the rest of the cascade. Accordingly, $R_i(a)$ is defined as average with respect to initial failures of deleted neighbors:

$$
\widetilde{R}_i(a) = \sum_{\mathbf{s}_{dnb(i)} \in \{0,1\}^{m_i}} R_i \left(a + \sum_{n \in dnb(i)} s_n \right) \prod_{n \in dnb(i)} p_{ni}^{s_n}(1 - p_{ni})^{1-s_n}.
$$

(4) Finally, the cascade size distribution is computed by SDP with inputs M and $\widetilde{R}_i$.

References

1. Newman, M. E. J., Strogatz, S. H. & Watts, D. J. Random graphs with arbitrary degree distributions and their applications. *Physcal Rev. E* **64**, 026118 (2001).
2. Friedman, N. *et al.* Universal critical dynamics in high resolution neuronal avalanche data. *Phys. Rev. Lett.* **108**, 208102, https://doi.org/10.1103/PhysRevLett.108.208102 (2012).
3. Dea, R., Crofts, J. J. & Kaiser, M. Spreading dynamics on spatially constrained complex brain networks. *J. The Royal Soc. Interface* **10**, https://doi.org/10.1098/rsif.2013.0016 (2013).
4. Newman, M. E. J. Spread of epidemic disease on networks. *Phys. Rev. E* **66**, 016128, https://doi.org/10.1103/PhysRevE.66.016128 (2002).
5. Battiston, S., Delli Gatti, D., Gallegati, M., Greenwald, B. C. N. & Stiglitz, J. E. Credit Default Cascades: When Does Risk Diversification Increase Stability? *J. Financial Stab.* **8**, 138–149 (2012).
6. Watts, D. J. A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci. USA* **99**, 5766–5771 (2002).
7. Burkholz, R., Garas, A. & Schweitzer, F. How damage diversification can reduce systemic risk. *Phys. Rev. E* **93**, 042313, https://doi.org/10.1103/PhysRevE.93.042313 (2016).
8. Burkholz, R., Leduc, M. V., Garas, A. & Schweitzer, F. Systemic risk in multiplex networks with asymmetric coupling and threshold feedback. *Phys. D: Nonlinear Phenom.* **323–324**, 64–72, http://www.sciencedirect.com/science/article/pii/S0167278915001943, https://doi.org/10.1016/j.physd.2015.10.004. Nonlinear Dynamics on Interconnected Networks (2016).
9. Burkholz, R. & Schweitzer, F. Framework for cascade size calculations on random networks. *Phys. Rev. E* **97**, 042312, https://doi.org/10.1103/PhysRevE.97.042312 (2018).
10. Burkholz, R. & Schweitzer, F. Correlations between thresholds and degrees: An analytic approach to model attacks and failure cascades. *Phys. Rev. E* **98**, 022306, https://doi.org/10.1103/PhysRevE.98.022306 (2018).
11. Bianconi, G. Rare events and discontinuous percolation transitions. *Phys. Rev. E* **97**, 022314, https://doi.org/10.1103/PhysRevE.97.022314 (2018).
12. Braunstein, A., Mézard, M. & Zecchina, R. Survey propagation: An algorithm for satisfiability. *Random Struct. & Algorithms* **27**, 201–226, https://doi.org/10.1002/rsa.20057 (2005).
13. Burkholz, R., Herrmann, H. J. & Schweitzer, F. Explicit size distributions of failure cascades redefine systemic risk on finite networks. *Sci. Reports*, 1–8, https://rdcu.be/NbZe, https://doi.org/10.1038/s41598-018-25211-3 (2018).
14. Friedli, S. & Velenik, Y. *Statistical Mechanics of Lattice Systems: A Concrete Mathematical Introduction* (Cambridge University Press, 2017).
15. Poledna, S. & Thurner, S. Elimination of systemic risk in financial networks by means of a systemic risk transaction tax. *Quant. Finance* **16**, 1599–1613, https://doi.org/10.1080/14697688.2016.1156146 (2016).
16. Gleeson, J. P. & Porter, M. A. Complex spreading phenomena in social systems. In Jørgensen, S. & Ahn, Y.-Y. (eds) *Complex Spreading Phenomena in Social Systems: Influence and Contagion in Real-World Social Networks*, 81–95, https://doi.org/10.1007/978-3-319-77332-2 (Springer, Cham, 2018).
17. Gleeson, J. P. Binary-state dynamics on complex networks: Pair approximation and beyond. *Phys. Rev. X* **3**, 021004, https://doi.org/10.1103/PhysRevX.3.021004 (2013).
18. Watts, D. J. & Dodds, P. S. Threshold models of social influence. In *The Oxford Handbook of Analytical Sociology*, chap. 20, 475–497 (Oxford University Press, Oxford, UK, 2009).
19. Granovetter, M. S. Threshold Models of Collective Behavior. *The Am. J. Sociol.* **83**, 1420–1443 (1978).
20. Kermack, W. O. & McKendrick, A. G. A contribution to the mathematical theory of epidemics. *Proc. Royal Soc. Lond. A: Math. Phys. Eng. Sci.* **115**, 700–721, http://rspa.royalsocietypublishing.org/content/115/772/700, https://doi.org/10.1098/rspa.1927.0118 (1927).
21. Kempe, D., Kleinberg, J. & Tardos, É. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD' 03, 137–146, https://doi.org/10.1145/956750.956769 (ACM, New York, NY, USA, 2003).
22. Jordan, M. I. (ed.) *Learning in Graphical Models* (MIT Press, Cambridge, MA, USA, 1999).
23. Norlen, K., Lucas, G., Gebbie, M. & Chuang, J. EVA: Extraction, Visualization and Analysis of the Telecommunications and Media Ownership Network. *Proc. Int. Telecommun. Soc. 14th Biennial Conf.* (*ITS2002*), Seoul Korea (2002).
24. Gleeson, J. P. & Cahalane, D. Seed size strongly affects cascades on random networks. *Phys. Rev. E* **75**, 1–4, https://doi.org/10.1103/PhysRevE.75.056103 (2007).
25. Altarelli, F., Braunstein, A., Dall'Asta, L., Wakeling, J. R. & Zecchina, R. Containing epidemic outbreaks by message-passing techniques. *Phys. Rev. X* **4**, 021024, https://doi.org/10.1103/PhysRevX.4.021024 (2014).
26. Lokhov, A. Y. Reconstructing parameters of spreading models from partial observations. In *Advances in Neural Information Processing Systems 29, Dec 5–10, Barcelona, Spain*, 3459–3467, http://papers.nips.cc/paper/6129-reconstructing-parameters-of-spreading-models-from-partial-observations (2016).
27. Barabási, A. L. & Albert, R. Emergence of Scaling in Random Networks. *Sci.* **286**, 509–512, https://doi.org/10.1126/science.286.5439.509 (1999).
28. Clauset, A., Shalizi, C. R. & Newman, M. E. J. Power-Law Distributions in Empirical Data. *Soc. for Ind. Appl. Math. Rev.* **51**, 661, https://doi.org/10.1137/070710111, http://link.aip.org/link/SIREAD/v51/i4/p661/s1{&}Agg=doi (2009).

Acknowledgements

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-019-42873-9.

Competing Interests: The author declares no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.