

Cascade Size Distributions: Why They Matter and How to Compute Them Efficiently

Rebekka Burkholz,¹ John Quackenbush^{1, 2}

¹ Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA 02115

² Harvard Medical School, Boston, MA 02115
rburkholz@hsph.harvard.edu, johnq@hsph.harvard.edu

Abstract

Cascade models are central to understanding, predicting, and controlling epidemic spreading and information propagation. Related optimization, including influence maximization, model parameter inference, or the development of vaccination strategies, relies heavily on sampling from a model. This is either inefficient or inaccurate. As alternative, we present an efficient message passing algorithm that computes the probability distribution of the cascade size for the Independent Cascade Model on weighted directed networks and generalizations. Our approach is exact on trees but can be applied to any network topology. It approximates locally tree-like networks well, scales to large networks, and can lead to surprisingly good performance on more dense networks, as we also exemplify on real world data.

Introduction

The Independent Cascade Model (ICM) is a cornerstone in the study of spreading processes on networks. It has been proven useful in the source detection of epidemic outbreaks (Leskovec et al. 2007; Farajtabar et al. 2015; Xu and Chen 2015; Zhu, Chen, and Ying 2017), classification of fake news (Tschitschek et al. 2018; Vosoughi, Roy, and Aral 2018), marketing (Leskovec, Adamic, and Huberman 2007; Kempe, Kleinberg, and Tardos 2005), or identification of causal miRNAs for cancer (Nalluri et al. 2017). It can be mapped to the SIR (Susceptible Infected Recovered) model (Kermack and McKendrick 1927), which together with its variants has served during the COVID-19 pandemic to estimate associated risks in different scenarios (Chinazzi et al. 2020; Zhang et al. 2020). Due to its simplicity, the model is well suitable to estimate a spreading process in situations of high uncertainty. It still provides mechanistic insights that allow to predict changes in the spreading dynamics due to different interventions to an ongoing cascade.

The main quantity of interest is the number of infected people, which is also called the cascade size. Finding a feasible policy that minimizes the risk of large cascade sizes poses a challenging optimization problem. It also requires to estimate model parameters based on highly uncertain data. Furthermore, the only information available is usually the number of infected people, in particular, at early stages of

the disease spreading. Another common optimization problem related to the ICM is influence maximization, i.e., the maximization of the average cascade size by optimal selection of seeds (i.e., initial spreaders) (Kempe, Kleinberg, and Tardos 2003). These exemplary optimization problems have in common that they require sampling from the probability distribution of the cascade size or variational inference approaches. Yet, sampling is computationally costly or inaccurate. Variational inference usually relies on Belief Propagation (Lokhov, Mézard, and Zdeborová 2015; Lokhov 2016), which estimates marginals in graphical models, i.e., the infection probability of each node in a network and their average. Yet, it cannot capture the strong positive dependence between node activations that is due to mutual infections. For similar reasons, the average cascade size can be a bad representation of the probability distribution, which is often broad and multi-modal (Burkholz, Herrmann, and Schweitzer 2018; Burkholz 2019), see also Fig. 4.

We propose an accurate and efficient alternative that is based on message passing like Belief Propagation but follows different principles, which we term Subtree Distribution Propagation (SDP). Subtree Distribution Propagation generalizes to discrete models with more than two states and can capture many SIR-type spreading processes. In contrast to Belief Propagation, we provide information on joint activations by computing the full probability distribution of the final cascade size (instead of the average cascade size). In addition, we obtain the infection probabilities conditional on the cascade size (instead of the unconditional ones).

Up to our knowledge, conditional infection probabilities have not been studied before but provide rich information about the spreading behavior and susceptibility of nodes. 1) They allow to identify nodes that are functionally similar with respect to the cascade process. These similarities are usually caused by network symmetries. In a biological setting, these can imply redundant pathways and are particularly relevant for deepening our understanding of diseases like cancer (Weighill et al. 2021). 2) Hence, they define a natural node embedding where the embedding space corresponds to the final cascade size distribution. Note that it is common practice to compare node embeddings by using them to estimate the infection probabilities of nodes in SIR-type spreading processes. As we show, these probabilities can also be used to define a node embedding directly.

3) Identifying node similarities (for instance by node clustering) has also algorithmic implications. They reduce the set of seed candidates in influence maximization or related problems and thus speed up Greedy approaches. In some cases, they can even enable exhaustive search (see supplementary material).

As Belief Propagation, the proposed algorithms are exact on trees. They require maximally $O(N^2)$ computations, where N denotes the number of network nodes, but the run time scales usually as $O(N)$ on sparse networks. Parallelization can speed this up further. As extension to general networks, we propose Tree Distribution Approximation (TDA). It combines Belief Propagation (BP) with Subtree Distribution Propagation (SDP). TDA is approximate, but accurate on locally tree-like networks (like BP). Beyond performance gains, our algorithms have the advantage over sampling that they provide us with functional relationships between cascade model parameters and outputs, i.e. the cascade size distribution and conditional infection probabilities. These would allow us to also compute gradients with respect to model parameters efficiently and thus enable first-order (instead of zeroth-order) optimization approaches.

The Independent Cascade Model

The ICM models the binary, stochastic, and discrete activation dynamics of nodes in an undirected network $G = (V, E)$ consisting of $N = |V|$ nodes that are connected by links in E . Each node i is either inactive ($s_i = 0$) or active ($s_i = 1$) and can only switch from an inactive to an active state, but not vice versa. Initially, each node i activates with probability p_i independently of the other nodes. In the next time step ($t = 1$), an active node i can trigger new activations of its neighbors $j \in \text{nb}(i) := \{j \mid (i, j) \in E\}$. Its degree $d_i = |\text{nb}(i)|$ counts the number of neighbors it has. Each neighbor j activates independently with probability w_{ij} and can cause new activations in the next time step. This way, a cascade keeps propagating, where several activations can happen at the same time t and each node becoming active at t can trigger new activations only in the next time step $t + 1$ but not any later times. The process ends at time $T \leq N$, when no further activations can occur. Then, the fraction of active nodes $\rho = 1/N \sum_{i=1}^N s_i(T)$ defines the final cascade size. This is the realization of a random variable C with probability distribution $p_C(\rho)$, as the cascade process is stochastic. $p_C(\rho)$ is also termed probability mass function of C and has support $\{0, 1/N, \dots, 1\}$. In summary, an ICM is parametrized by (\mathbf{p}, \mathbf{W}) , where the vector \mathbf{p} has components p_i and the matrix \mathbf{W} entries w_{ij} . Note that W is often related to network weights and can encode directedness by $w_{ij} \neq w_{ji}$ and $w_{ij} = 0$.

Contributions

Our main contribution is the development of efficient message passing algorithms that compute the final cascade size distribution and node infection probabilities conditional on the final cascade size for the Independent Cascade Model. Subtree Distribution Propagation (SDP) is exact on trees and requires $O(N^2)$ computations. Tree Distribution Propagation (TDA) applies to general networks and leads to accurate

inference of the cascade size distribution on locally tree-like networks. Additional backpropagation computes the infection probabilities of nodes conditional on the final cascade size. As we show in experiments, our algorithms scale favorably with increasing network size and lead to speed-ups by a factor ranging from 60 – 500 in comparison with sampling. Efficiency gains are not only relevant for modeling spreading phenomena on large networks but also on smaller ones if the cascade size distribution is required repeatedly for different parameters, as it is common in most related optimization approaches. We further investigate the limitations of our algorithms by studying them on random graphs that are not locally tree-like but find surprisingly good performance. In addition, we provide examples on real world data sets, including a dense network of miRNA signaling corresponding to gastrointestinal cancer and a large YouTube network.

Related Literature

As diverse as spreading phenomena are the related optimization objectives. The insight that the average cascade size is a submodular influence function (Kempe, Kleinberg, and Tardos 2003) has inspired many efforts to maximize this quantity by nearly optimal seed size selection (Du et al. 2014; Y. Lokhov and Saad 2016) or network adjustments (Wen et al. 2017). This has great applications, e.g., in marketing (Morris 2000; Goldenberg, Libai, and Muller 2001; Domingos and Richardson 2001). Other works are more concerned with destructive aspects of cascades and their mitigation to avoid epidemic spreading (Budak, Agrawal, and El Abbadi 2011; Y. Lokhov and Saad 2016). Then, the not always explicit objective is to minimize cascades or to create boundary conditions that limit their size (Burkholz et al. 2016; Burkholz and Schweitzer 2018a, 2019, 2018b). In some cases, the objective can also be to keep cascades within a specified range (Burkholz and Dubatovka 2019; Noël, Brummitt, and D’Souza 2013). Another example for a related optimization problem are maximum likelihood approaches to infer cascade model parameters in the context of information propagation in social networks. Several works assume the knowledge of full (Myers and Leskovec 2010; Gomez-Rodriguez, Balduzzi, and Schölkopf 2011; Du et al. 2012) or at least partial (Lokhov 2016; Abrahao et al. 2013) diffusion information, i.e., data on which node becomes infected when, while the network is mostly known or has to be learned (Hoffmann and Caramanis 2019; Gripon and Rabbat 2013). Likelihoods are analytically attainable when the time evolution is observed. When this information is missing but node identities are known, network recovery is sometimes still feasible (Amin, Heidari, and Kearns 2014). However, this kind of data is usually not available during an epidemic outbreak. Hence, epidemic spreading models are mainly calibrated by parameter grid search and model sampling (Chinazzi et al. 2020; Zhang et al. 2020) and most of the related optimization approaches are based on average cascade sizes and/or sampling from the cascade model.

To compute the average cascade size, faster alternatives to sampling are provided by Local Tree Approximations (Newman 2002) for large random networks or Belief Propagation for smaller sparse networks (Gleeson and Porter 2018;

Y. Lohkov and Saad 2016; Burkholz 2019). For specific optimization problems like influence maximization, also more efficient sampling techniques have been developed, which reuse samples or terminate non-promising simulations in smart ways. Prominent examples are given by RIS (Borgs et al. 2014) and SSA (Nguyen, Thai, and Dinh 2016) sampling.

Yet, as recently shown (Cheng et al. 2014; Burkholz, Herrmann, and Schweitzer 2018; Burkholz 2019), average cascade sizes do not provide relevant summary statistics in cases when the underlying cascade size distributions are broad, multi-modal, and support extreme events. Examples are provided in Fig. 4. In every case, the full probability distribution provides much richer information about a network structure. It is also necessary for defining the likelihood of observed cascades and related model estimation. In addition, it allows to estimate the risk of extreme events. Usually, those events are of highest interest to judge the robustness of a system or to find optima (Battiston et al. 2016; McNeil, Frey, and Embrechts 2015; Ohsaka and Yoshida 2017). For instance, (Ohsaka and Yoshida 2017) maximizes the expected shortfall of the final cascade size with respect to a portfolio of seeds by sampling from an ICM. Often, very large or small cascades occur only with small probabilities. This makes the optimization of tails harder, in particular, by sampling and demands alternatives. For a given simple ICM with uniform infection probability $p_i = p$ or threshold model and a locally tree-like network, the final cascade size distribution can be computed by message passing (Burkholz 2019) in $O(N^2 \log(N))$. Our approach is inspired, even though the math is different and our algorithms are more efficient ($O(N^2)$). Furthermore, we can capture general ICMs with heterogeneous weights and initial activation probabilities, while (Burkholz 2019) is restricted to the uniform case $p_i = w_{ij} = p$. In addition, we provide activation probabilities of nodes conditional on the final cascade size.

Motivation: Why Cascade Size Distributions?

To motivate the benefits of taking cascade size distributions into account, let us consider a toy example of a social network, as depicted in Fig. 1, which is exposed to an epidemic that evolves according to an ICM. Without any interventions, the cascade size is distributed as visualized by the gray histogram. The black dashed line marks the average cascade size, which does not correspond to the most probable events and does not prepare policy makers for the high risk of large cascades.

Next, we assume that we can mitigate the risk with vaccinations. Our goal is to avoid large cascades that would exceed the available hospital capacities and to minimize the number of deaths. Who should be vaccinated first? Nodes 1 and 2 are the best candidates with identical exposure $\sum_j w_{ij}$. Node 1 interacts with less people but has closer contact with them than Node 2. Interestingly, vaccinating 1, the node with the smaller degree, would minimize the risk of large cascades $\mathbb{P}(\varrho \geq 0.75)$, while vaccinating 2 would minimize the average cascade. If hospital capacities are limited,

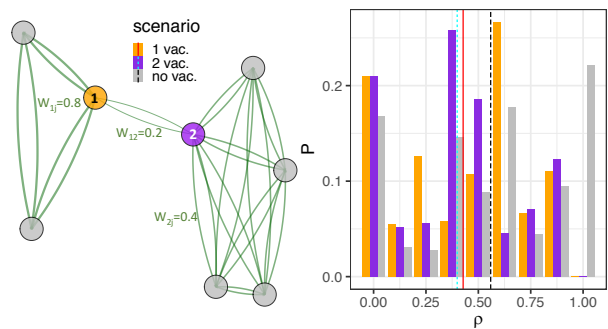


Figure 1: Comparing the average cascade size and tail risk. The histograms show the cascade size distribution in different scenarios for an ICM with $p_i = 0.2$, $w_{12} = w_{21} = 0.2$, $w_{ij} = 0.8$ within the clique of size 3, and $w_{ij} = 0.4$ within the clique of size 5. Lines mark average cascade sizes. Vaccinating Node 2 minimizes the average cascade size, while vaccinating Node 1 minimizes the probability of large cascades $\mathbb{P}(\varrho \geq 0.75)$.

vaccinating 1 might be the most promising strategy, which requires the estimation of the cascade size distribution’s tail.

Algorithmic Approach

Next, we describe the novel message passing algorithms that we propose to estimate the final cascade size distribution and conditional activation probabilities of nodes. While we focus in our derivations on the ICM, the principle is in fact more general and could be transferred to continuous time (Saito et al. 2009) and different binary graphical models or more complicated epidemic spreading processes, in which nodes are equipped with a higher number of discrete states.

Message Passing Algorithms

Fig. 2 gives an overview of our main contribution: four variants of a message passing algorithm. The core is formed by *Subtree Distribution Propagation (SDP)*, which computes the final cascade size distribution based on a rooted tree and ICM as input. Every node is visited only once and the algorithm can be parallelized according to the tree structure. Starting in the leaves (i.e., nodes with degree $d_n = 1$), each node n sends information about the cascade size distribution of the subtree T_n rooted in n to its parent p . It only requires information by its children as input. Finally, the output is constructed in the root. The relevant variables are visualized in Fig. 3 C.

To compute the activation probability of nodes conditional on the final cascade size, *Conditional Subtree Distribution Propagation (conSDP)* adds a backpropagation routine to SDP. Information is sent from the root to its children until it reaches the leaves. Thus, each node is visited twice in total.

SDP and conSDP are exact, but are limited to trees. To obtain an approximate variant that applies to any (simple) network G , we introduce (*conditional*) *Tree Distribution Approximation (TDA)* as extension of (conditional) SDP. The

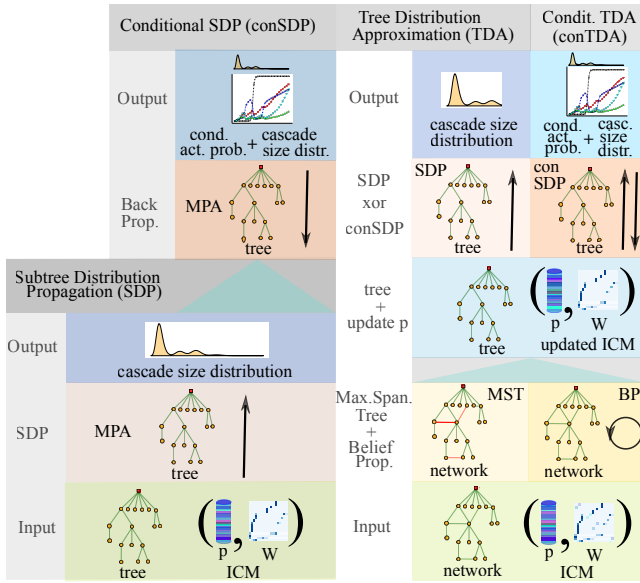


Figure 2: Overview of different message passing algorithm variants for the final cascade size distribution. All are exact on trees, where SDP coincides with TDA and ConSDP with ConTDA. TDA and conTDA are approximative for networks that are different from trees. In addition, conSDP or conTDA variants compute conditional activation probabilities of nodes.

goal is to reduce G to a tree M (for instance a maximum spanning tree (MST)) and run SDP (or conSDP) on M . However, to compensate for the removal of edges and thus dependencies of node activations, we increase the initial activation probabilities \mathbf{p} of the ICM. There, we assume that former neighbors j have activated independently before a node i with a probability p_{ji} that we estimate by Belief Propagation (BP). If BP does not converge, we could substitute it by another approach such as the junction tree algorithm. For simplicity and computational efficiency, we will only consider BP. In the following, we detail the information propagation equations of the respective algorithms.

Subtree Distribution Propagation

Intuition The derivation of SDP relies solely on combinatorial arguments respecting the right order of activations. This order is encoded in messages that a node n sends to its parent p about the number of active nodes a_n in the subtree T_n , which is rooted in n (see Fig. 3 C). Messages of type A assume that p activates (either before or after the node n), while messages of type B consider the case that p does not activate (at least before n). We need to distinguish these two cases because they enable us to combine the messages received by children efficiently.

We would like to compute the probability distribution of the number $a_n = \sum_{i \in T_n} s_i$ of active nodes in T_n . This is given by the sum of n 's state s_n and the number of active nodes a_{c_i} in the subtrees that are rooted in n 's k children c_i : $a_n = s_n + \sum_{i=1}^k a_{c_i}$. The challenge is that the distributions

of the summands depend on each other. Yet, before the activation of n , the subtrees a_{c_i} are independent. The activation of n has an impact on the subtrees but also independently of each other. Hence, in both cases, the probability distribution of their sum can be computed efficiently as convolution of the right distributions associated with a_{c_i} . The distributions of a_{c_i} correspond to messages of type A in case that n becomes active and messages of type B in case that n does not. In case of the activation of n , we have to distinguish two message subtypes, i.e., A^Σ and A^0 . A^0 is only auxiliary to subtract the probability of an infeasible case, in which n is not activated by any of its children.

The messages that n receives by its children are then used to compute the distributions of a_n for different cases. p_{na} refers to “no activation” of n , p_{la} to “late activation” (or “no initial activation”) of n , and p_a to all cases when n activates. p_{la} is auxiliary to subtract from p_a the case that no child successfully triggers the activation of n . At the end of the message passing, the cascade size distribution is computed in the root, where no parent state needs to be considered. In case of a final activations, either the root does not activate with $p_{na}(a)$ or activates with $p_a(a-1)$ so that $a-1$ other nodes become active. The precise algorithm is stated in the following theorem.

Theorem 1 (SDP). *The final cascade size distribution $p_C(\rho)$ of an ICM (\mathbf{p}, \mathbf{W}) on a tree G with root r and N nodes is given as output of the following message passing algorithm. Starting in the leaves, each node n sends the functions $p_{B_n}(a)$, $p_{A_n^0}(a)$, and $p_{A_n^\Sigma}(a)$ for $a = 0, \dots, N$ as messages to its parent p . We have*

$$\begin{aligned} p_{B_n}(0) &= 1 - p_n, & p_{B_n}(1) &= p_n(1 - w_{np}), \\ p_{A_n^0}(0) &= (1 - p_n)(1 - w_{pn}), & p_{A_n^0}(1) &= (1 - p_n)w_{pn} \\ & & & + p_n(1 - w_{np}), \\ p_{A_n^\Sigma}(0) &= (1 - p_n)(1 - w_{pn}), & p_{A_n^\Sigma}(1) &= p_{A_n^0}(1) + p_n w_{np} \end{aligned}$$

for a leaf n (with degree $d_n = 1$). Otherwise, define for a node n with k children c_1, \dots, c_k :

$$\begin{aligned} p_{na}(a) &= (1 - p_n)p_{B_{c_1}} * \dots * p_{B_{c_k}}[a], \\ p_{la}(a) &= (1 - p_n)p_{A_{c_1}^0} * \dots * p_{A_{c_k}^0}[a], \\ p_a(a) &= p_{A_{c_1}^\Sigma} * \dots * p_{A_{c_k}^\Sigma}[a] - p_{la}(a), \end{aligned}$$

where $*$ denotes a convolution. An intermediate node $n \neq r$ ($d_n > 1$) with $k_n = d_n - 1$ children sends the messages:

$$\begin{aligned} p_{A_n^0}(a) &= (1 - w_{pn})p_{na}(a) + w_{pn}p_{la}(a-1) \\ &+ (1 - w_{np})p_a(a-1), \quad p_{B_n}(a) = p_{na}(a) + (1 - w_{np}) \\ &\times p_a(a-1), \quad p_{A_n^\Sigma}(a) = p_{A_n^0}(a) + w_{np}p_a(a-1). \end{aligned}$$

In the root, $n = r$ with $k_r = d_r$ children, the final cascade size distribution is given as: $p_C(a/N) = p_{na}(a) + p_a(a-1)$.

Note that all operations can be performed in Fourier space. Thus, the convolutions simplify to elementwise multiplication of vectors. The exact algorithm in Fourier space and a proof are provided in the supplement.

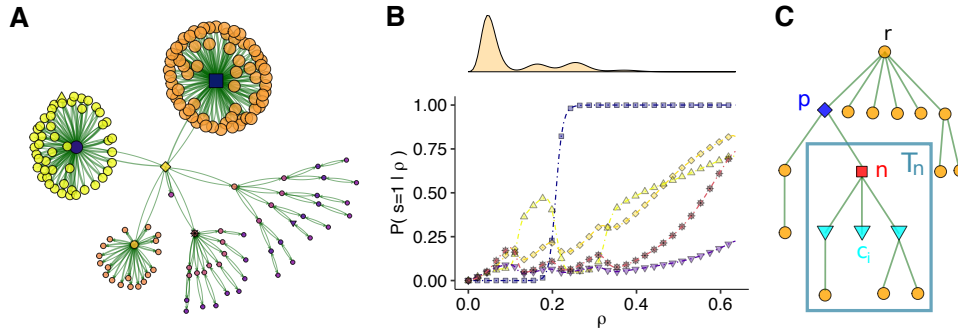


Figure 3: A. Exemplary tree with SD parameters. Link strengths are proportional to the ICM weights, node sizes to the activation probabilities. Node colors indicate cluster membership obtained from conditional activation probabilities. B. Conditional activation probability of nodes matching the symbols in A corresponding to the cascade size distribution on top. C. Illustration of variables in SDP.

Tree Distribution Approximation

To apply the same principle to a general network $G = (V, E)$ and thus SDP to a spanning tree $M = (V_M, E_M)$ of G , we have to regard the direct influence of neighbors $\text{dn}(i) = \{j \in V \mid (i, j) \in E, (i, j) \notin E_M\}$ on a node i , which are not connected with i in M any more. For all nodes, those can be estimated by Belief Propagation (BP).

Belief Propagation for the ICM Given an ICM (\mathbf{p}, \mathbf{W}) on a network G , the probability $p_{ij} = \mathbb{P}(s_i = 1 \mid s_j = 0)$ that i activates without j 's contribution is estimated as

$$Q_i(t+1) = (1 - p_i) \prod_{n \in \text{nb}(i)} (1 - w_{ni} p_{ni}(t)),$$

$$p_{ij}(t+1) = 1 - \frac{Q_i(t+1)}{1 - w_{ji} p_{ji}(t)}, \quad i, j = 1, \dots, N,$$

over $t = 0, \dots, R$ iteration steps. We initialize with the initial ICM activation probability $p_{ij}(0) = p_i$. Note that these equations are novel on their own and have been independently derived by Lokhov et al. (Lokhov and Saad 2019), who prove that they are exact on trees and that the iteration steps t correspond to the time of the cascade process.

Next, $p_{ij} = p_{ij}(R)$ for $j \in \text{dn}(i)$ are used to adapt the initial activation probabilities $\mathbf{p}^{(M)}$ of an ICM on the spanning tree M (instead of G). They incorporate the influence of deleted neighbors by assuming that they activate independently before i with p_{ji} .

ICM on MST We define an ICM $(\mathbf{p}^{(M)}, \mathbf{W}^{(M)})$ on M :

$$p_i^{(M)} = 1 - (1 - p_i) \prod_{j \in \text{dn}(i)} (1 - w_{ji} p_{ji}), \quad w_{ij}^{(M)} = w_{ij}$$

for $i = 1, \dots, N$ and $(i, j) \in E_M$.

Conditional Activation Probabilities

The activation probability of a node conditional on the final cascade size is straight forward to compute for the root at the end of SDP (see Thm. 1) as $\mathbb{P}(s_r = 1 \mid C = a/N) = p_a(a-1)/p_C(a/N)$. Yet, to obtain the same for every other

node n (that we turn into a root), we have to calculate additional messages that parents send back to their children. After SDP, only messages from the former parent are missing, so that $p_{B_p^c}(a)$, $p_{A_p^{oc}}(a)$, and $p_{A_p^{\Sigma c}}(a)$, where p is treated as a child, while n is the new parent. Thus, starting in the root r , each parent p backpropagates messages to their children n , where $\mathbb{P}(s_n = 1 \mid C = a/N)$ can be computed.

BackPropagation for ConSDP (or ConTDA) Using the notation of Thm. 1, each parent p sends to its child n :

$$p_{A_p^{oc}}(a) = (1 - w_{np}) p_{af \setminus n}^{(p)}(a) + w_{np} p_{la \setminus n}^{(p)}(a - 1)$$

$$+ (1 - w_{pn}) p_{a \setminus n}^{(p)}(a - 1), \quad p_{B_p^c}(a) = p_{af \setminus n}^{(p)}[a] + (1 - w_{pn})$$

$$\times p_{a \setminus n}^{(p)}(a - 1), \quad p_{A_p^{\Sigma c}}(a) = p_{A_p^{oc}}(a) + w_{np} p_{a \setminus n}^{(p)}(a - 1),$$

where the contribution of n in the convolution forming $p_x^{(p)}$ is removed (for $x \in \{af, lf, f\}$). Note the swap of n and p in comparison with Thm. 1. For all neighbors, children and parent, the messages from SDP are combined with the new one received by the parent as:

$$p_{na}^{(n)}(a) = p_{na} * p_{B_p^c}[a], \quad p_{la}^{(n)}(a) = p_{la} * p_{A_p^{oc}}[a],$$

$$p_a^{(n)}(a) = (p_a + p_{la}) * p_{A_p^{\Sigma c}}[a] - p_{la}^{(n)}(a)$$

and form the conditional activation probability of n as $\mathbb{P}(s_n = 1 \mid C = a/N) = p_a^{(n)}(a - 1) / (p_{na}^{(n)}(a) + p_a^{(n)}(a - 1))$. The precise algorithm is stated in the supplementary material.

Algorithmic Complexity

To validate our claim that our algorithms are usually more efficient than sampling, we have to make sure that we compare with the fastest available sampler. The precise algorithm is straight forward and stated in the supplement. Formally, its algorithmic complexity is of the same order as our algorithms for sparse networks, i.e. $O(N^2)$, but the omitted constant factors differ considerably.

SDP. In SDP, each node n is visited once, where mainly the convolution of $d_n - 1$ distributions of size $N + 1$ in

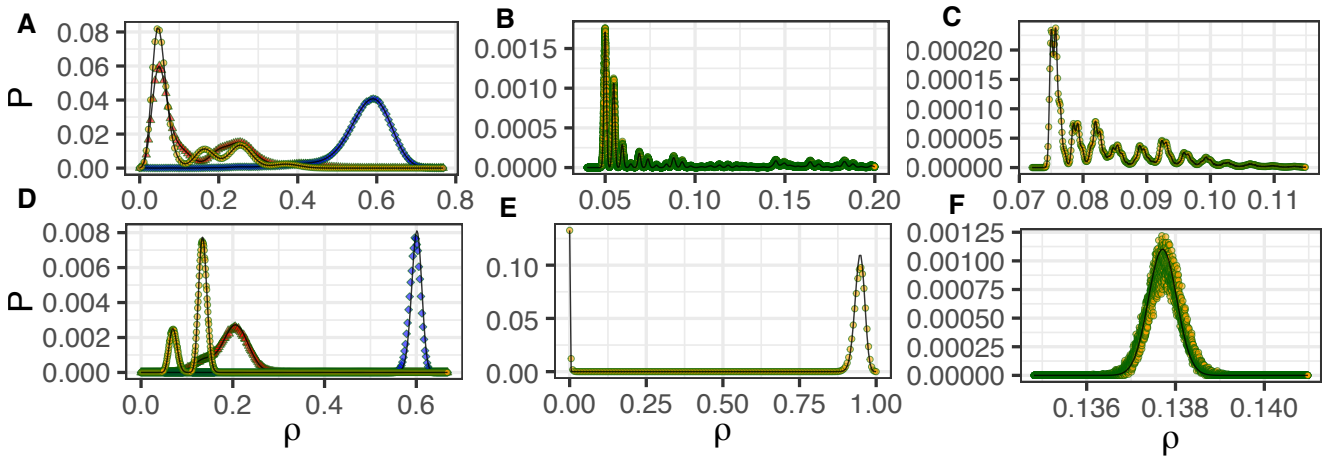


Figure 4: ainal cascade size distribution for TDA (lines) and simulations (symbols). SD (orange circles), DD (blue squares), ED (red triangles). A. Tree ($N = 181$). B. Tree ($N = 10^5$) with HD. C. Tree ($N = 10^6$) with HD. D. Corporate ownership network ($N = 4475$). E. Cancer miRNA network with empirical weights ($N = 201$). F. YouTube network ($N = 1134890$).

Fourier space has to be computed. This can be achieved with $O(d_n N)$ operations. In total, this adds up to $O(N \sum_n d_n) = O(N^2)$ operations, as $\sum_n d_n = 2 * (N - 1)$ in a tree. Back-propagation requires $O(N^2)$ operations for similar reasons.

TDA. TDA consists of two additional pre-processing steps: (1) Belief Propagation and (2) the computation of a spanning tree. (1) BP is linear in the number of edges $O(m)$ and is thus maximally of the same complexity as SDP (because $m \leq N^2$). (2) A random spanning tree suffices and can be constructed in $O(m)$. Finding a maximum spanning tree with Kruskal for optimal approximation quality could take longer for dense networks, i.e. $m \log(N)$.

In practice, for sparse networks and small number of nodes with large degrees, TDA is usually of order $O(N)$, since we can compute the convolutions of distributions with smaller support than N on subtrees. Alternatively, one could think of an approximate version of SDP that computes $p_C(\rho)$ for a finite resolution (for instance on an equidistant grid of $[0, 1]$). This could reduce also the worst case complexity to $O(N)$. With parallelization of computations in nodes that have received the messages by their children, this could be even brought down to $O(h)$, meaning that it is linear in the height of the input tree.

Sampling. Each sample of a cascade size can be obtained in $O(m)$ operations. Hence, the total run time is of order $O(NS)$, where S refers to the number of samples. How large should S be? To approximate the average cascade size, $S = 10^4$ is common (Kempe, Kleinberg, and Tardos 2003) and we compare TDA with this choice in Fig. 6 C. TDA achieves a speed-up by a factor of 60 – 500. Yet, depending on the cascade size distribution, 10^4 samples can lead to poor approximation results. The number of required samples is higher, since we have to estimate N free parameters and most of these parameters are small and scale as $1/N$. To obtain good enough estimates of each probability, we therefore need at least $O(N)$ samples, where the constant depends on

the approximation error. This results in a total complexity of $O(mN)$, which becomes $O(N^2)$ for sparse networks. According to our experiments, the constant still has to be relatively large.

Numerical Experiments

We perform detailed experiments on networks with different properties like size, average degree, and clustering coefficient. All computations were conducted on a MacBook Pro with 2.9 GHz Intel Core i9 processor and 32 GB 2400 MHz DDR4 memory. Fig. 4 provides an overview. The first row corresponds to trees of different size, while the second row shows cascade size distributions for real world networks. The first is a locally tree-like network of corporate ownership relationships (Norlen et al. 2002) with $N = 4475$ nodes, the second a dense correlation network of miRNA expression profiles using data from gastrointestinal cancer (Nalluri et al. 2017) with $N = 201$ nodes, and the third is a YouTube friendship network (Yang and Leskovec 2012) consisting of $N = 1134890$ nodes. ICM parameters are varied to demonstrate the diversity of possible cascade size distributions. Usually, we assume that all nodes activate initially with probability $p_i = 0.05$. For miRNAs, $p_i = 0.01$ to mitigate extensive spreading due to the high network density. Unless stated otherwise, we fix the weight parameters to a social dynamics (SD) model with $w_{ij} = 0.05 + 0.5d_i/d_j/Z$ and $Z = \max_{i,j}(d_i/d_j)$, in which hubs (i.e. nodes with high degrees) are more likely to activate network neighbors but are less likely to become activated by nodes with smaller degrees. For variation, we also report results for the following models: Financial systemic risk models are often concerned with different risk diversification mechanisms (Battiston et al. 2016; Burkholz, Garas, and Schweitzer 2016). Exposure Diversification (ED) $w_{ij} = 0.05 + 0.5/d_j$ assumes that high degree nodes are difficult to activate by single network

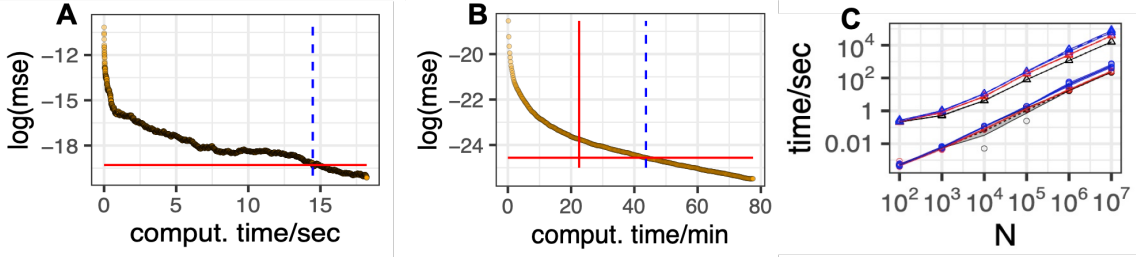


Figure 5: Run time comparisons. A&B. Mean squared error (mse) of sampling after indicated computation time for SD parameters. The ground truth is defined based on 10^6 samples. Red horizontal line: mse achieved by TDA. Red vertical line: run time of TDA. Blue dotted line: Time at which sampling achieves the same mse as TDA. A. Tree ($N = 181$). TDA takes 0.00078 seconds. B. YouTube network ($N = 1134890$). C. 10 independent ER-networks of size N are created with average degree $\bar{d} = 2.5$ (black), $\bar{d} = 4$ (red), $\bar{d} = 10$ (blue). Cascade model parameters are drawn iid as $p_n \sim U[1/N, 10/N]$ and $w_{np} \sim U[0.05, 1]$. The run time of sampling the cascade size 10^4 times (triangles) is compared with TDA (circles) on the same network. Lines refer to averages over 10 networks and shaded areas to the respective 0.95 confidence region. On average, TDA is faster by a factor ranging from 68 to 485.

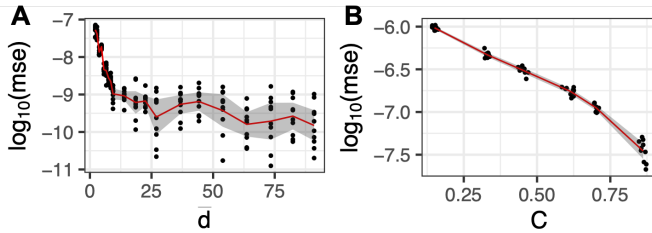


Figure 6: Approximation quality of TDA displayed as logarithmic mean squared error (mse) compared with the result of 10^6 independent Monte Carlo simulations. Each data point corresponds to one experiment. The red line refers to the mean of 10 experiments and the gray area to the associated 0.95 confidence interval. A. Trend with respect to increasing mean degree \bar{d} of locally tree-like random graphs. B. Trend with respect to increasing clustering coefficient of a random graph.

neighbors, while in damage diversification (DD) models high degree nodes less likely to infect a network neighbor, for instance, reflected by the choice $w_{ij} = 0.6$ for $d_i \geq d_j$ and $w_{ij} = 0.8$. The Hub diversification (HD) model is similar but makes hubs more difficult to activate with $w_{ij} = 0.5/d_j$ for $d_i < 0.1d_{max}$ and $w_{ij} = 0.4$ otherwise. Last but not least, we also consider models, in which the ICM parameters are drawn from random distributions.

Cascade size distributions are relevant. Our first observation based on Fig. 4 is that the displayed cascade size distributions are often broad and multi-modal. This can also be the case for large networks. For instance, Fig. 4 B&C belong to networks of size $N = 10^5$ and $N = 10^6$ respectively. Visually, we also find good agreement between SDP or TDA and sampling for 10^6 samples. To study this in more detail we first focus on trees, for which our algorithms are provably exact and provide us with ground truth results. These allow us to estimate the number of

required samples for Monte Carlo (MC) simulations to converge and, in comparison, to study the run time and scaling of our proposed algorithms.

How many samples are required? The answer must depend on the desired approximation quality and the shape of the cascade size distribution. Fig. 5 A shows the evolution of the mean squared error for an increasing number of samples for the small tree with corresponding SD distribution shown in Fig. 4 A. TDA and sampling become indistinguishable with 554800 or more samples. In comparison, the SD distribution for the YouTube network is more concentrated. Hence, already 28300 samples achieve a similar mean squared error as TDA (see Fig. 5 B). Yet, TDA is 18790 times faster in case of the tree, and almost 2 times faster in case of the YouTube network.

Scaling of TDA. Without prior knowledge of properties of the cascade size distribution, it is common practice to fix the amount of samples. In this case, sampling has a run-time complexity of $O(N)$, while TDA has formally a complexity of $O(N^2)$. Yet, TDA also scales as $O(N)$ for sparse networks with $m = O(N)$ and bounded degrees. This is demonstrated in Fig. 5 C, where we compare the run time for increasing network size N and sampling with $S = 10^4$ samples. $S = 10^4$ is a common choice to estimate the average cascade size. While this is often not sufficient to estimate the cascade size distribution or the probability of rare events, TDA is still faster by factors ranging from 68 – 485. We also note that TDA scales to large networks consisting of $N = 10^7$ nodes and needs less than 5 minutes to complete in this case.

Approximation quality of TDA. For trees, SDP is exact. For locally tree-like structures, i.e. networks with small clustering coefficient and thus negligible number of short loops, we expect that TDA approximates the cascade size distribution well. Fig. 4 also shows excellent agreement between TDA and sampling with $S = 10^6$ for the corporate ownership network and, surprisingly, even for the dense cancer

network. The YouTube network, however, is a difficult case. BP fails to provide a good estimates of p_{ij} and overestimates the average cascade size. Hence, we have to define $p_{ij} = p_i$ in TDA and estimate the average cascade size based on a few (i.e. 100) samples. All reported results on the YouTube network are for an accordingly shifted cascade size distribution that meets this estimated average. This combination of TDA and a few sampling steps is still much faster than sampling alone as shown in Fig. 5 B. To analyze systematically, how deviations from a locally tree-like network structure influence the approximation quality of TDA, we present Fig. 6. Specifically, we study the effect of increasing average degree and thus higher network density and increased clustering. Surprisingly, TDA leads to relatively good approximation results. A high average degree is less problematic as long as the network has a small number of triangles or short cycles. Also a relatively high clustering coefficient does not hamper the approximation results as long as the average degree is small. We vary this coefficient in Fig. 5 B when we generate random graphs consisting of $N = 1000$ nodes according to (Schank and Wagner 2005), while we try to keep the average degree fixed. The exact parameter choices are detailed in the supplementary material.

Conditional activation probabilities. The conditional activation probabilities of exemplary nodes of the small tree (Fig. 3A) are shown in Fig. 3B, as obtained by ConSDP. Corresponding figures for the other networks can be found in the supplement. Conditional activation probabilities vary substantially with the cascade size and often increase non-monotonically, different from what might be expected. They contain rich information not only about the probability of a node to activate but also about its role in the spreading process. Big hubs are more predictable and are always active above a certain cascade size. Their activation usually marks larger cascades but does not explain the largest. These only occur with the activation of nodes that are more difficult to reach by cascades. Nodes, which are topological interchangeable, also have an identical conditional activation probability. The identification of such symmetries is particularly interesting in the analysis of biological networks like the cancer related miRNA network, since these hint towards similar functions of nodes within pathways and thus redundancies in the network. In addition, similar conditional activation probabilities translate into similar effects as seeds. Therefore, we can interpret the conditional probabilities as node embedding which associates each node i with a vector v^i with components $v_{r+1}^i = \mathbb{P}(s_i = 1 \mid \rho = r/N)$. Any clustering algorithm could be employed for a dimensionality reduction based on v_i . For simplicity, we choose kmeans to obtain 15 clusters. Node colors in Fig. 3 indicate the cluster membership.

Discussion

The core algorithms that we have derived, Subtree Distribution Propagation (SDP) and conSDP, compute the final cascade size distribution and conditional activation probabilities given a general independent cascade model (ICM) and can therefore replace expensive sampling procedures in related optimization problems. Furthermore, they provide a

functional relationship between model parameters and algorithmic output. On their basis, efficient algorithms that compute gradients can be derived. This can enable first order optimization approaches in cases in which only zeroth order optimization is available so far.

In addition to the cascade size distribution, we can compute the activation probabilities of nodes conditional on the final cascade size. These are particularly informative in systemic risk analyses, as they allow the focus on extreme events. In addition, they provide a node embedding that allows to cluster nodes with similar functionality for the cascade process.

While our algorithms are exact on trees, real world networks usually have additional connections. Tree Distribution Propagation (TDA) provides excellent approximation results for locally tree-like networks.

However, if networks consist of multiple short loops, these loops introduce stronger dependencies of activations than TDA can capture. Highly connected nodes activate all together with higher probability. By approximating a denser network with a tree in TDA, we treat some nodes as conditionally independent when they are not and thus underestimate the variance of the cascade size distribution. In this sense, we can interpret TDA as variational approach to obtain a proxy for the final cascade size distribution, where TDA captures more dependencies than variational approaches based on BP alone could provide.

Future work could improve such approximations by combining SDP with sampling. SDP only requires estimates of cascade size distributions for subgraphs (considering states of parents), which could also be sampled. As long as such subgraphs are connected like trees, SDP can be used to combine the distributions. Such an approach could speed up sampling approaches and improve the accuracy of TDA in highly clustered networks.

Our algorithms could also be extended to allow for distributions on the ICM parameters and could therefore aid robust influence maximization under model parameter uncertainty (Kalimeris, Kaplun, and Singer 2019) or Bayesian model learning approaches.

Code Availability

The code in R, Python, and C++ is available on GitHub: <https://github.com/rebekka-burkholz/TDA>

Supplements

Supplementary data with proofs and more details about the presented algorithms is available at <https://arxiv.org/abs/1909.05416>.

Acknowledgements

RB and JQ were supported by a grant from the US National Cancer Institute (1R35CA220523).

We thank Alkis Gotovos for helpful feedback on the manuscript.

References

- Abrahamo, B.; Chierichetti, F.; Kleinberg, R.; and Panconesi, A. 2013. Trace Complexity of Network Inference. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13*, 491–499.
- Amin, K.; Heidari, H.; and Kearns, M. 2014. Learning from Contagion (without Timestamps). In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, II-1845-II-1853*.
- Battiston, S.; Caldarelli, G.; May, R. M.; Roukny, T.; and Stiglitz, J. E. 2016. The price of complexity in financial networks. *Proceedings of the National Academy of Sciences* 113(36): 10031–10036.
- Borgs, C.; Brautbar, M.; Chayes, J.; and Lucier, B. 2014. Maximizing Social Influence in Nearly Optimal Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14*, 946–957.
- Budak, C.; Agrawal, D.; and El Abbadi, A. 2011. Limiting the Spread of Misinformation in Social Networks. In *Proceedings of the 20th International Conference on World Wide Web, WWW '11*, 665–674. ISBN 978-1-4503-0632-4.
- Burkholz, R. 2019. Efficient message passing for cascade size distributions. *Scientific Reports* 5: 6561.
- Burkholz, R.; and Dubatovka, A. 2019. Initialization of ReLUs for Dynamical Isometry. In *Advances in Neural Information Processing Systems* 32, 2385–2395. NeurIPS'2019.
- Burkholz, R.; Garas, A.; and Schweitzer, F. 2016. How damage diversification can reduce systemic risk. *Physical Review E* 93: 042313.
- Burkholz, R.; Herrmann, H. J.; and Schweitzer, F. 2018. Explicit size distributions of failure cascades redefine systemic risk on finite networks. *Scientific Reports* 1–8.
- Burkholz, R.; Leduc, M. V.; Garas, A.; and Schweitzer, F. 2016. Systemic risk in multiplex networks with asymmetric coupling and threshold feedback. *Physica D: Nonlinear Phenomena* 323–324: 64 – 72.
- Burkholz, R.; and Schweitzer, F. 2018a. Correlations between thresholds and degrees: An analytic approach to model attacks and failure cascades. *Physical Review E* 98: 022306.
- Burkholz, R.; and Schweitzer, F. 2018b. Framework for cascade size calculations on random networks. *Physical Review E* 97: 042312.
- Burkholz, R.; and Schweitzer, F. 2019. International crop trade networks: The impact of shocks and cascades. *Environmental Research Letters* 14: 11.
- Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can Cascades Be Predicted? In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, 925–936.
- Chinazzi, M.; Davis, J.; Ajelli, M.; Gioannini, C.; Litvinova, M.; Merler, S.; Pastore, Y.; Piontti, A.; Mu, K.; Rossi, L.; Sun, K.; Viboud, C.; Xiong, X.; Yu, H.; Halloran, M.; Longini, I. J.; and Vespignani, A. 2020. A simple model of global cascades on random networks. *Science* 368(6489): 395–400.
- Domingos, P.; and Richardson, M. 2001. Mining the Network Value of Customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01*, 57–66.
- Du, N.; Liang, Y.; Balcan, M.; and Song, L. 2014. Influence Function Learning in Information Diffusion Networks. In Xing, E. P.; and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 2016–2024. Beijing, China.
- Du, N.; Song, L.; Yuan, M.; and Smola, A. J. 2012. Learning Networks of Heterogeneous Influence. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems* 25, 2780–2788. NIPS'2012.
- Farajtabar, M.; Gomez-Rodriguez, M.; Zamani, M.; Du, N.; Zha, H.; and Song, L. 2015. Back to the Past: Source Identification in Diffusion Networks from Partially Observed Cascades. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*.
- Gleeson, J. P.; and Porter, M. A. 2018. Complex Spreading Phenomena in Social Systems. In Jørgensen, S.; and Ahn, Y.-Y., eds., *Complex Spreading Phenomena in Social Systems: Influence and Contagion in Real-World Social Networks*, 81–95. Springer.
- Goldenberg, J.; Libai, B.; and Muller, E. 2001. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters* 12(3): 211–223.
- Gomez-Rodriguez, M.; Balduzzi, D.; and Schölkopf, B. 2011. Uncovering the Temporal Dynamics of Diffusion Networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 561–568. ICML'11.
- Gripon, V.; and Rabbat, M. 2013. Reconstructing a Graph from Path Traces. *CoRR* abs/1301.6916.
- Hoffmann, J. H.; and Caramanis, C. 2019. Learning Graphs from Noisy Epidemic Cascades. *Joint International Conference on Measurement and Modeling of Computer Systems* .
- Kalimeris, D.; Kaplun, G.; and Singer, Y. 2019. Robust Influence Maximization for Hyperparametric Models. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 3192–3200.
- Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. 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. New York, NY, USA.
- Kempe, D.; Kleinberg, J.; and Tardos, E. 2005. Influential Nodes in a Diffusion Model for Social Networks. In

- Proceedings of the 32Nd International Conference on Automata, Languages and Programming, ICALP'05*, 1127–1138.
- Kermack, W. O.; and McKendrick, A. G. 1927. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 115(772): 700–721.
- Leskovec, J.; Adamic, L. A.; and Huberman, B. A. 2007. The Dynamics of Viral Marketing. *ACM Trans. Web* 1(1).
- Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective Outbreak Detection in Networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07*, 420–429.
- Lokhov, A. Y. 2016. Reconstructing Parameters of Spreading Models from Partial Observations. In *Advances in Neural Information Processing Systems 29, Dec 5-10, Barcelona, Spain*, 3459–3467.
- Lokhov, A. Y.; Mézard, M.; and Zdeborová, L. 2015. Dynamic message-passing equations for models with unidirectional dynamics. *Physical Review E* 91: 012811.
- Lokhov, A. Y.; and Saad, D. 2019. Scalable Influence Estimation Without Sampling. *arXiv* 1912.12749.
- McNeil, A. J.; Frey, R.; and Embrechts, P. 2015. *Quantitative Risk Management: Concepts, Techniques and Tools Revised edition*, volume 1. Princeton University Press, 2 edition.
- Morris, S. 2000. Contagion. *The Review of Economic Studies* 67(1): 57–78. URL <http://dx.doi.org/10.1111/1467-937X.00121>.
- Myers, S.; and Leskovec, J. 2010. On the Convexity of Latent Social Network Inference. In *Advances in Neural Information Processing Systems 23*, 1741–1749. NIPS'2010.
- Nalluri, J.; Rana, P.; Barh, D.; Azevedo, V.; Dinh, T.; Vladimirov, V.; and Ghosh, P. 2017. Determining causal miRNAs and their signaling cascade in diseases using an influence diffusion model. *Scientific Reports* 7.
- Newman, M. E. J. 2002. Spread of epidemic disease on networks. *Physical Review E* 66: 016128.
- Nguyen, H. T.; Thai, M. T.; and Dinh, T. N. 2016. Stop-and-Stare: Optimal Sampling Algorithms for Viral Marketing in Billion-Scale Networks. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD '16*, 695–710.
- Noël, P.-A.; Brummitt, C. D.; and D'Souza, R. M. 2013. Controlling Self-Organizing Dynamics on Networks Using Models that Self-Organize. *Phys. Rev. Lett.* 111: 078701. doi:10.1103/PhysRevLett.111.078701. URL <https://link.aps.org/doi/10.1103/PhysRevLett.111.078701>.
- Norlen, K.; Lucas, G.; Gebbie, M.; and Chuang, J. 2002. EVA: Extraction, Visualization and Analysis of the Telecommunications and Media Ownership Network. *Proceedings of International Telecommunications Society 14th Biennial Conference (ITS2002), Seoul Korea*.
- Ohsaka, N.; and Yoshida, Y. 2017. Portfolio Optimization for Influence Spread. In *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, 977–985.
- Saito, K.; Kimura, M.; Ohara, K.; and Motoda, H. 2009. Fearing Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis. In *Advances in Machine Learning 2009*, volume 5828 of *ACML'2009*.
- Schank, T.; and Wagner, D. 2005. Approximating Clustering Coefficient and Transitivity. *J. Graph Algorithms Appl.* 9(2): 265–275.
- Tschiatschek, S.; Singla, A.; Gomez Rodriguez, M.; Merchant, A.; and Krause, A. 2018. Fake News Detection in Social Networks via Crowd Signals. In *Companion Proceedings of the The Web Conference 2018, WWW '18*, 517–524.
- Vosoughi, S.; Roy, D.; and Aral, S. 2018. The spread of true and false news online. *Science* 359(6380): 1146–1151.
- Weighill, D.; Guebila, M. B.; Lopes-Ramos, C.; ; Glass, K.; Quackenbush, J.; Platig, J.; and Burkholz, R. 2021. Gene Regulatory Network Inference as Relaxed Graph Matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI'2021.
- Wen, Z.; Kveton, B.; Valko, M.; and Vaswani, S. 2017. Online Influence Maximization under Independent Cascade Model with Semi-Bandit Feedback. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 3022–3032. NIPS'2017.
- Xu, W.; and Chen, H. 2015. Scalable Rumor Source Detection under Independent Cascade Model in Online Social Networks. In *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 236–242.
- Y. Lokhov, A.; and Saad, D. 2016. Optimal Deployment of Resources for Maximizing Impact in Spreading Processes. *Proceedings of the National Academy of Sciences* 114.
- Yang, J.; and Leskovec, J. 2012. Defining and Evaluating Network Communities Based on Ground-Truth. In *2012 IEEE 12th International Conference on Data Mining*, 745–754.
- Zhang, J.; Litvinova, M.; Liang, Y.; Wang, Y.; Wang, W.; Zhao, S.; Wu, Q.; Merler, S.; Viboud, C.; Vespignani, A.; Ajelli, M.; and Yu, H. 2020. Changes in contact patterns shape the dynamics of the COVID-19 outbreak in China. *Science*.
- Zhu, K.; Chen, Z.; and Ying, L. 2017. Catch'Em All: Locating Multiple Diffusion Sources in Networks with Partial Observations. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 1676–1683.