Mapping flows on sparse networks with missing links

Jelena Smiljanić,[1,2,*] Daniel Edler,[1,3,4] and Martin Rosvall[1]

[1]*Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden*
[2]*Scientific Computing Laboratory, Center for the Study of Complex Systems, Institute of Physics Belgrade, University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia*
[3]*Gothenburg Global Biodiversity Centre, Box 461, SE-405 30 Gothenburg, Sweden*
[4]*Department of Biological and Environmental Sciences, University of Gothenburg, Carl Skottsbergs gata 22B, Gothenburg 41319, Sweden*

(Received 12 December 2019; revised 12 May 2020; accepted 9 June 2020; published 6 July 2020)

Unreliable network data can cause community-detection methods to overfit and highlight spurious structures with misleading information about the organization and function of complex systems. Here we show how to detect significant flow-based communities in sparse networks with missing links using the map equation. Since the map equation builds on Shannon entropy estimation, it assumes complete data such that analyzing undersampled networks can lead to overfitting. To overcome this problem, we incorporate a Bayesian approach with assumptions about network uncertainties into the map equation framework. Results in both synthetic and real-world networks show that the Bayesian estimate of the map equation provides a principled approach to revealing significant structures in undersampled networks.

DOI: 10.1103/PhysRevE.102.012302

I. INTRODUCTION

Unraveling the modular organization of social and biological systems with interactions comprising measured movements of some entity such as people, money, or information requires reliable maps of network flows [1–5]. To find modular regularities in network flows, the map equation estimates a modular description length of the flows with information-theoretic measures. Optimizing the map equation with the search algorithm Infomap maximally compresses the modular description and detects significant flow-based communities when enough links are observed [2,6]. However, if too many links are missing, then the map equation may highlight spurious communities resulting from mere noise. While there are generative methods that can deal with uncertain network structures, including link-prediction algorithms [7–9] and network reconstruction approaches that often build on the stochastic block model [10–14], no method can reliably identify flow-based communities in networks with missing links.

The map equation estimates the modular description length of network flows with the Shannon entropy [15]. With missing data, the Shannon entropy underestimates the actual entropy of the complete data [16]. Consequently, when a network has many missing links, the map equation underestimates the actual description length of the complete network, capitalizes on details in the observed network, and favors network partitions with many small communities. While higher model complexity can further compress the description length, the resulting communities become sensitive to network perturbations. Having more missing links further obscures the community structure and leads to higher sensitivity. Overfitting happens when the communities poorly compress the description length of the complete network or other samples of the complete network [17,18].

Underestimating the entropy in networks with missing links also causes problems for standard procedures that evaluate model-prediction performance, including cross-validation: When the modular description length depends on the number of observed links, it also depends on the number of cross-validation folds such that only balanced but wasteful equal-sized splits of a network into training and test networks give useful results.

To overcome these problems, we present two regularization methods based on entropy estimation for undersampled discrete data. First, we incorporate a Bayesian approach in the map equation framework [19] and derive a closed-form formula for the posterior mean of the map equation under the Dirichlet prior distribution of network flows. Second, to enable more effective cross-validation, we measure the modular description length of the training and test networks for a given partition using Grassberger entropy estimation [20].

We show that the Bayesian estimate of the map equation does not detect spurious communities in the undersampled regime in either synthetic or real-world networks. Also, compared with the degree-corrected stochastic block model [21,22], this approach gives solutions that are more robust to missing links in the analyzed networks. Moreover, with Grassberger entropy estimation, the modular description length becomes nearly independent of the amount of data: Instead of wasteful equal-sized splits, we can use most links in

[*] jelena.smiljanic@umu.se

the training network to detect communities with Infomap and validate them using the remaining links in the test network. These two complementary solutions help us reduce overfitting and allow us to detect significant flow-based communities in networks with missing links.

II. MAPPING FLOWS ON COMPLETE NETWORKS

The map equation is an information-theoretic objective function for community detection based on the equivalence between data compression and identifying regularities in data. Building on this minimum description length principle, the map equation estimates the per-step theoretical lower limit of the average code word length needed to describe network flows with a modular description [2,6]. When the links themselves do not represent flows, we can model the network flows with a random walker traversing the network. The goal is to identify the network partition that maximally compresses the modular description, which, at the same time, best captures the modular regularities of the network flows.

For simplicity, here we consider modular descriptions with a two-level community hierarchy (for the multilevel map equation, see Appendix B). In a network with a well-defined community structure, the network flows stay for a relatively long time within communities. Therefore, to encode movements of the random walker between nodes with better compression, the map equation reuses short code words in modular codebooks instead of using unique code words for each node. For a uniquely decodable description, this approach requires an additional index codebook to encode transitions between communities.

The map equation measures the theoretical lower limit of the code length using the Shannon entropy [15]. For partition M of nodes $\alpha = 1 \dots V$ in communities $i = 1 \dots m$, the map equation takes as input the probability that the random walker enters community i, $q_{i\curvearrowright}$, the probability to visit node α, p_α, and the probability to exit community i, $q_{i\curvearrowleft}$. With $p_i^{\circlearrowright} = q_{i\curvearrowleft} + \sum_{\alpha\in i} p_\alpha$ for the total use rate of module codebook i, the average per-step code length needed to describe random walker movements within community i is

$$H(\mathcal{P}_i) = -\frac{q_{i\curvearrowleft}}{p_i^{\circlearrowright}} \log_2 \frac{q_{i\curvearrowleft}}{p_i^{\circlearrowright}} - \sum_{\alpha\in i} \frac{p_\alpha}{p_i^{\circlearrowright}} \log_2 \frac{p_\alpha}{p_i^{\circlearrowright}}. \qquad (1)$$

Similarly, the average per-step code length needed to describe random walker transitions between communities is

$$H(\mathcal{Q}) = -\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{q_{\curvearrowright}} \log_2 \frac{q_{i\curvearrowright}}{q_{\curvearrowright}}, \qquad (2)$$

where $q_{\curvearrowright} = \sum_{i=1}^{m} q_{i\curvearrowright}$ is the total use rate of the index codebook. Therefore, we can express the map equation as the sum of the average code length of all codebooks weighted by their use rate:

$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p_i^{\circlearrowright} H(\mathcal{P}_i). \qquad (3)$$

To identify the partition that minimizes the map equation, Infomap explores the space of possible solutions in a stochastic and greedy fashion.

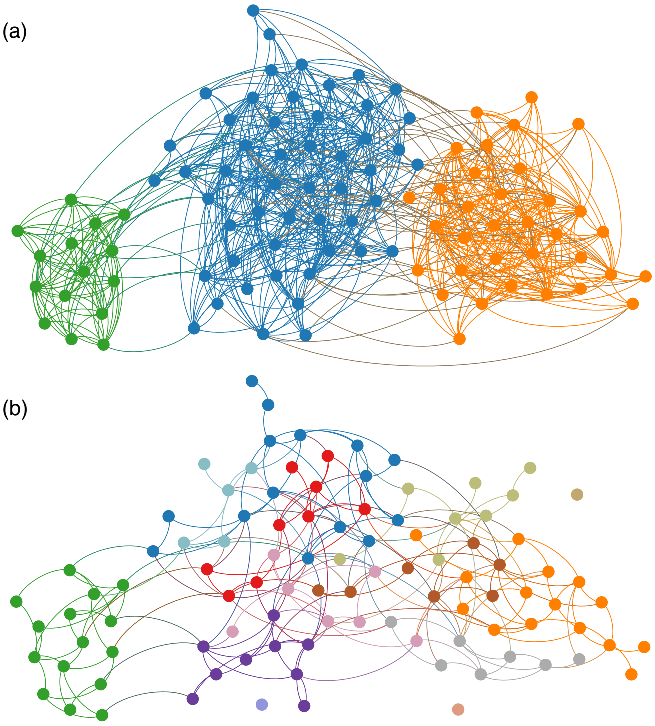

FIG. 1. Illustration of the overfitting problem in a small modular network. (a) The network has three communities. (b) When observing only a fraction of the links, the identified thirteen communities misrepresent the underlying network structure.

III. MAPPING FLOWS ON SPARSE NETWORKS WITH MISSING LINKS

Combined with Infomap, the map equation is an accurate method for community detection when complete network data are available [23]. However, empirical network data can lack data or contain measurement errors that cause missing or spurious links. When the map equation is applied to such unreliable network data, it may identify spurious communities with misleading information about the underlying network structure and function (Fig. 1).

We focus on missing links, a common problem in social and biological networks, that causes the sample estimates of the random walker's transition probabilities to lose precision. When plugging the estimates into the Shannon entropy, the obtained entropy estimator suffers from a negative bias and underestimates the entropy terms of the map equation [16]. Consequently, for the same partition M, the description length decreases and the relative code length savings over the one-module solution, $l = 1 - L(\mathsf{M})/L(\mathsf{1})$, increases with the number of missing links (Fig. 2).

Worse yet, underestimating the index and module codebooks distorts their balance and shifts the optimal solution. The index codebook underrates the increase in between-module description length when using more communities, and the module codebooks overrate the within-module compression gain when using smaller communities. Also, stochastic fluctuations in missing links can lead the search algorithm off track because more undersampled regions attract community boundaries. Capitalizing on noise in this way underestimates

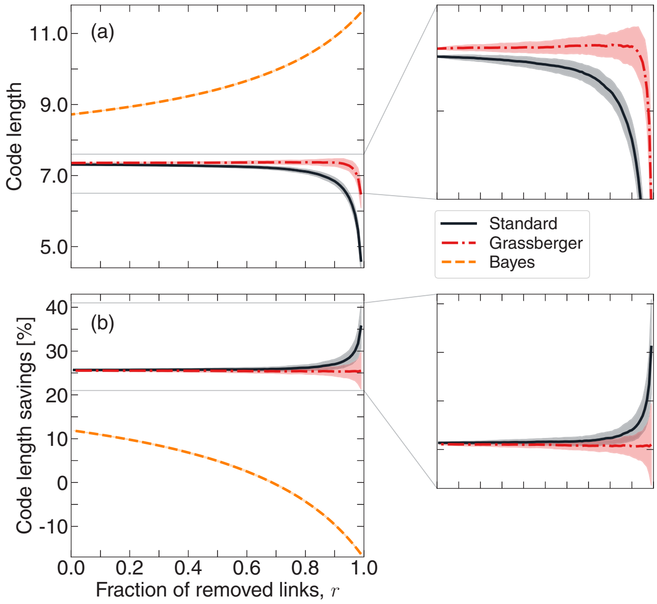

FIG. 2. Modular compression in sparse networks. (a) Modular code length for planted partitions after link removal. (b) Relative code length savings in networks for planted partitions. With standard entropy estimation, the average description length decreases as the number of missing links increases, and we cannot compare relative code length savings in networks with different densities. In contrast, Grassberger entropy estimation almost eliminates the code length's density dependency. For $r > 0.7$, the code length savings are negative for the Bayesian estimate of the map equation with prior $a_\alpha = \ln(V)$. By preferring the one-module solution over the planted partition in severely undersampled networks, the Bayesian estimate of the map equation avoids overfitting. For each r, we plot averages and variances over 100 network samplings of the synthetic network described in Sec. IV.

not only the codebooks but also, primarily, the transition rates between communities. As a result, the map equation favors more and smaller communities in sparse networks with missing links [9] (Fig. 1). This effect is evident when so many links are missing that actual communities become sparse or even form disconnected components. Then the map equation cannot detect the actual communities; instead it overfits and identifies spurious communities from mere noise in the network. To overcome overfitting, we incorporate a Bayesian estimate of the map equation.

A. Bayesian estimate of the map equation

Different methods have been proposed to address the problem of entropy underestimation [19,20,24–27]. Methods based on bias reduction cannot prevent overfitting of the map equation because they have a high variance in the undersampled regime [20,24,25] and cannot deal with the underestimation of the transition rates between communities. Instead, we use a Bayesian approach proposed by Wolpert and Wolf to estimate the function of probability distributions [19]. This method not only prevents overfitting to noisy structures better than other Bayesian estimators [26,27]; it also enables an analytical estimation of the map equation and a computationally efficient implementation in Infomap.

In general, we seek the Bayesian estimator $\hat{f}_B$ of a function $f(\boldsymbol{\rho})$ that takes a discrete probability distribution $\boldsymbol{\rho} = (\rho_1, \rho_2, \dots, \rho_m)$ as input. When $\boldsymbol{\rho}$ is not given and we have only observations $\boldsymbol{n} = (n_1, n_2, \dots, n_m)$, with $\sum_{i=1}^m n_i = N$ sampled according to the distribution $\boldsymbol{\rho}$ $(E(n_i) = \rho_i N)$, we must estimate $f(\boldsymbol{\rho})$ using the observed data $\boldsymbol{n}$. The Bayesian estimator for $f(\boldsymbol{\rho})$ is the posterior average,

$$\hat{f}_B(\boldsymbol{n}) = E[f|\boldsymbol{n}] = \int f(\boldsymbol{\rho})P(\boldsymbol{\rho}|\boldsymbol{n})d\boldsymbol{\rho}, \tag{4}$$

where $P(\boldsymbol{\rho}|\boldsymbol{n})$ is the posterior over the unknown distribution $\boldsymbol{\rho}$ given by Bayes' rule,

$$P(\boldsymbol{\rho}|\boldsymbol{n}) = \frac{P(\boldsymbol{n}|\boldsymbol{\rho})P(\boldsymbol{\rho})}{P(\boldsymbol{n})}. \tag{5}$$

To obtain $P(\boldsymbol{\rho}|\boldsymbol{n})$, we choose an appropriate prior probability distribution $P(\boldsymbol{\rho})$ and use the fact that the likelihood

$$P(\boldsymbol{n}|\boldsymbol{\rho}) = N! \prod_{i=1}^{m} \frac{\rho_i^{n_i}}{n_i!} \tag{6}$$

and the total probability of the data

$$P(\boldsymbol{n}) = \int d\boldsymbol{\rho} P(\boldsymbol{n}|\boldsymbol{\rho})P(\boldsymbol{\rho}). \tag{7}$$

Applied to the map equation, we seek the Bayesian estimator of $f(\boldsymbol{\rho}) = L(\mathsf{M})$. Assuming undirected and unweighted links, the transition rate estimates are [28]

$$p_\alpha = \frac{k_\alpha}{\sum_{\alpha=1}^V k_\alpha}, \tag{8}$$

$$q_{i\curvearrowleft} = \frac{k_{i\curvearrowleft}}{\sum_{\alpha=1}^V k_\alpha}, \tag{9}$$

$$q_{i\curvearrowright} = \frac{k_{i\curvearrowright}}{\sum_{\alpha=1}^V k_\alpha}, \tag{10}$$

where k_α is the degree of node α and $k_{i\curvearrowleft} = k_{i\curvearrowright}$ is the degree of module i, the number of links that connect nodes of module i with nodes of other modules j, $j \neq i$. However, when the information about links is incomplete, the actual values of node and module degrees can deviate from these estimates. Therefore, we must apply a probabilistic approach, or the map equation will overfit and exploit spurious network structures.

To develop a Bayesian treatment of the map equation, for a given partition M, we specify a prior distribution $P(p_\alpha, q_{i\curvearrowleft}, q_{i\curvearrowright})$ over the transition rates p_α, $q_{i\curvearrowleft}$, and $q_{i\curvearrowright}$. A convenient choice is the Dirichlet distribution, which has simple analytical properties and can be interpreted as a probability distribution over the multinomial distribution of the transition rates,

$$P(p_\alpha, q_{i\curvearrowleft}, q_{i\curvearrowright}|a_\alpha, a_{i\curvearrowleft}, a_{i\curvearrowright}) = \frac{\Gamma(a_1 + \dots + a_{m\curvearrowright})}{\Gamma(a_1)\dots\Gamma(a_{m\curvearrowright})} \prod_{\alpha=1}^{V} p_\alpha^{a_\alpha - 1} \prod_{i=1}^{m} q_{i\curvearrowleft}^{a_{i\curvearrowleft}-1} \prod_{i=1}^{m} q_{i\curvearrowright}^{a_{i\curvearrowright}-1}. \tag{11}$$

Here $\Gamma(x)$ is the gamma function and $a_1, \dots, a_V$, $a_{1\curvearrowleft}, \dots, a_{m\curvearrowleft}$, and $a_{1\curvearrowright}, \dots, a_{m\curvearrowright}$ are the parameters of the distribution. While $\sum_{\alpha=1}^V p_\alpha + \sum_{i=1}^m q_{i\curvearrowleft} + \sum_{i=1}^m q_{i\curvearrowright} \neq 1$, we can use normalized transition rates because the map equation is scale invariant (see Appendix A).

We obtain the posterior distribution of the transition rates in Eq. (5) by multiplying the Dirichlet prior by the likelihood function and normalizing:

$$\begin{aligned} &P(p_\alpha, q_{i\curvearrowleft}, q_{i\curvearrowright} | k_\alpha, k_{i\curvearrowleft}, k_{i\curvearrowright}, a_\alpha, a_{i\curvearrowleft}, a_{i\curvearrowright}) \\ &\propto \prod_{\alpha=1}^{V} p_\alpha^{k_\alpha + a_\alpha - 1} \prod_{i=1}^{m} q_{i\curvearrowleft}^{k_{i\curvearrowleft} + a_{i\curvearrowleft} - 1} \prod_{i=1}^{m} q_{i\curvearrowright}^{k_{i\curvearrowright} + a_{i\curvearrowright} - 1}. \end{aligned} \tag{12}$$

By combining this distribution and the expanded form of the map equation,

$$\begin{aligned} L(\mathsf{M}) = &- \sum_{\alpha=1}^{V} p_\alpha \log_2(p_\alpha) - \sum_{i=1}^{m} q_{i\curvearrowright} \log_2(q_{i\curvearrowright}) \\ &+ \sum_{i=1}^{m} \left(q_{i\curvearrowright} + \sum_{\alpha \in i} p_\alpha \right) \log_2 \left(q_{i\curvearrowright} + \sum_{\alpha \in i} p_\alpha \right) \\ &- \sum_{i=1}^{m} q_{i\curvearrowleft} \log_2(q_{i\curvearrowleft}) \\ &+ \left(\sum_{i=1}^{m} q_{i\curvearrowleft} \right) \log_2 \left(\sum_{i=1}^{m} q_{i\curvearrowleft} \right), \end{aligned} \tag{13}$$

in Eq. (4), and integrating, we obtain a closed formula for the posterior average of the map equation,

$$\begin{aligned} \hat{L}_B(\mathsf{M}) = &\frac{1}{\ln(2)} \frac{1}{\sum_{\alpha=1}^{V} u_\alpha} \\ &\times \Bigg[- \sum_{\alpha=1}^{V} u_\alpha \psi(u_\alpha + 1) - \sum_{i=1}^{m} u_{i\curvearrowright} \psi(u_{i\curvearrowright} + 1) \\ &+ \sum_{i=1}^{m} \left(u_{i\curvearrowright} + \sum_{\alpha \in i} u_\alpha \right) \psi \left(u_{i\curvearrowright} + \sum_{\alpha \in i} u_\alpha + 1 \right) \\ &- \sum_{i=1}^{m} u_{i\curvearrowleft} \psi(u_{i\curvearrowleft} + 1) \\ &+ \left(\sum_{i=1}^{m} u_{i\curvearrowleft} \right) \psi \left(\sum_{i=1}^{m} u_{i\curvearrowleft} + 1 \right) \Bigg], \end{aligned} \tag{14}$$

where $u_x = k_x + a_x$ and $\psi(x)$ is the digamma function.

The parameters $\boldsymbol{a}$ reflect our prior assumption of the link distribution in the network before we observed the network data. After seeing the data, we update our assumption by increasing the value of a_x by k_x and obtain the posterior distribution. For a sparse, undersampled network, therefore, the prior parameters $\boldsymbol{a}$ dominate the posterior link distribution. Conversely, as the network density increases, the posterior distribution becomes sharply peaked and the network data dominate the posterior link distribution. Proper selection of prior parameters $\boldsymbol{a}$ is important for good performance.

We consider as an uninformative prior an Erdős-Rényi network with V nodes, where each pair of nodes is connected with some constant probability p [29]. The average degree is $\langle k \rangle = pV$ and sets the prior parameters to $a_\alpha = \langle k \rangle$ and $a_{i\curvearrowleft} = a_{i\curvearrowright} = V_i(V - V_i)\frac{\langle k \rangle}{V-1}$, where V_i is the number of nodes in module i. We aim to choose the average degree $\langle k \rangle$ such that the prior prevents the map equation from overfitting in the undersampled network, but also enables the map equation to detect well-formed communities. Since the random network experiences a phase transition from disconnected to connected at $\langle k \rangle = \ln(V)$ [29], for $\langle k \rangle \ll \ln(V)$ the random network has isolated components and the prior cannot prevent overfitting, while for $\langle k \rangle \gg \ln(V)$ well-formed communities can merge such that the map equation underfits. At the phase transition between these extremes, $a \sim \ln(V)$ forms a principled prior.

Because there are no modular regularities in an Erdős-Rényi network, this choice of prior induces positive bias in the code length estimation [Fig. 2(a)]. When observing fewer links in a network, the prior network influences the posterior link distribution more such that the code length increases for the planted partition. Eventually, for severely undersampled networks, the Bayesian estimate of the map equation prefers the one-module solution and thereby avoids overfitting [Fig. 2(b)].

This Bayesian estimate of the map equation extends to weighted networks where complete information about link weights is missing. If the link weights represent flows such that no flow modeling is necessary, then the method also works for directed networks.

We have implemented the Bayesian estimate of the map equation in Infomap, available for anyone to use [30]. While we restrict our paper to the two-level formulation of the map equation for the sake of simplicity, the code also handles the Bayesian estimate of the multilevel map equation (see Appendix B).

B. The map equation with Grassberger entropy estimation

An informative comparison between the standard map equation and a map equation with corrected entropy terms must take into account the structural properties of the detected communities. When possible, we can compare detected communities with planted communities; however, this approach does not work for real networks without known communities. To test for under- or overfitting in any network, we use cross-validation.

We first split the network data into training and test sets and apply Infomap to identify the partition that maximally compresses the description length of the training network. If Infomap successfully recovers a significant partition of the training network, then the partition with maximal modular code length savings over the one-level code length will also successfully compress the description length of the test network. The opposite happens when there is not enough evidence in the data. Then Infomap overfits and detects a partition in the training network without code length savings in the test network. Thus, if Infomap detects a significant partition M without overfitting, the relative code length savings in the test network should be positive, $l^{\text{test}} = 1 - L_{\text{test}}(\mathsf{M})/L_{\text{test}}(\mathsf{1}) > 0$ and close to the relative code length savings of the training network, $l^{\text{test}} \sim l^{\text{train}}$. Conversely, if Infomap overfits we expect $l^{\text{test}} < 0$.

However, the fact that the description length and the relative code length savings vary with the fraction of observed links limits the choice of training and test networks (Fig. 2). Only with equal-sized training and test networks will the standard map equation underestimate their true description lengths to the same degree. But since equal splits waste half of the links on the test network, the training network of already sparse networks will be severely undersampled and possibly below the detectability limit. To reduce the description length's dependency on the fraction of observed links and enable effective cross-validation, we incorporate Grassberger entropy estimation [20] into the map equation.

For effective cross-validation, Grassberger entropy estimation enables the use of most of the links in the training network. We construct a test network by randomly removing a fraction r of links from the network. The remaining links form a training network. With E for the total number of links in the network and k_α for the degree of node α, the probability that k'_α links of node α remain in the training network after removing $E - E' = rE$ links follows the hypergeometric distribution:

$$P(k'_\alpha) = \frac{\binom{k_\alpha}{k'_\alpha}\binom{E-k_\alpha}{E'-k'_\alpha}}{\binom{E}{E'}}. \tag{15}$$

If E, E', and k_α are sufficiently large, then the hypergeometric distribution converges toward the Poisson distribution,

$$P(k'_\alpha) = \frac{\lambda^{k'_\alpha}}{k'_\alpha!}e^{-\lambda}, \tag{16}$$

where the parameter $\lambda = \frac{E'k_\alpha}{E} = (1-r)k_\alpha$ such that $\langle k'_\alpha\rangle = (1-r)k_\alpha$.

For a given incomplete set of observations $(n_1, n_2, \dots, n_m)$, Grassberger entropy estimation assumes that they come from Poisson distributions with mean values $(z_1, z_2, \dots, z_m)$ and aims to construct a function $\phi(n)$ that minimizes the error $|z_i \ln(z_i) - E[n_i\phi(n_i)]|$ across all values of z_i [20]. The solution that minimizes the error is a recursive function $\phi(n) = G_n$ defined as

$$G_1 = -\gamma - \ln(2), \quad G_2 = 2 - \gamma - \ln(2),$$
$$G_{2n+1} = G_{2n}, \quad G_{2n+2} = G_{2n} + \frac{2}{2n+1}, \tag{17}$$

where γ is Euler's constant [20].

While we cannot use Grassberger entropy estimation for weighted or directed networks, where visit rates correspond to the PageRank of the nodes [6], it does work for unweighted and undirected networks, where node visit and module transition rate estimates are given by link counts, Eqs. (8)–(10). Assuming incomplete observations, we can incorporate Grassberger entropy estimation into the map equation such that Eq. (13) takes the form

$$\begin{aligned}\hat{L}_G(\mathsf{M}) = {}& \frac{1}{\ln(2)}\frac{1}{\sum_{\alpha=1}^{V}k_\alpha} \\ &\times\Bigg[-\sum_{\alpha=1}^{V}k_\alpha G_{k_\alpha} - \sum_{i=1}^{m}k_{i\curvearrowright}G_{k_{i\curvearrowright}} \\ &+ \sum_{i=1}^{m}\Bigg(k_{i\curvearrowright} + \sum_{\alpha\in i}k_\alpha\Bigg)G_{k_{i\curvearrowright}+\sum_{\alpha\in i}k_\alpha} \\ &- \sum_{i=1}^{m}k_{i\curvearrowleft}G_{k_{i\curvearrowleft}} + \Bigg(\sum_{i=1}^{m}k_{i\curvearrowleft}\Bigg)G_{\sum_{i=1}^{m}k_{i\curvearrowleft}}\Bigg].\end{aligned} \tag{18}$$

Grassberger entropy estimation also works for the multilevel formulation of the map equation [31].

Grassberger entropy estimation has high variance and low bias [32]. Due to its high variance in the undersampled regime (Fig. 2) and its lack of prior that can deal with underestimating the transition rates between communities, the map equation with Grassberger entropy estimation paired with Infomap does not perform better than the standard map equation on sparse networks with missing links. However, thanks to its low bias, the map equation with Grassberger entropy estimation applied to cross-validation with averaged code length over several network samplings can dramatically reduce the code length dependency on network density [Fig. 2(a)]. Also, for planted partitions, the average relative code length savings is practically independent of network density [Fig. 2(b)]. Consequently, we can use most links in the training network to reliably detect communities with Infomap.

IV. RESULTS AND DISCUSSION

We first analyze a synthetic network with planted community structure and a real-world Jazz collaboration network [33]. We generate the synthetic network with the Lancichinetti-Fortunato-Radicchi (LFR) method [34]. It has $V = 1000$ nodes, average node degree $\langle k\rangle = 16$, and nodes partitioned into $M = 35$ communities. The mixing parameter $\mu = 0.3$ is the probability that a randomly chosen link will connect nodes from different communities. In the Jazz collaboration network, each node represents a band and two nodes are connected if there is at least one musician who has played in both bands. For this network with 198 nodes and 2742 links, there is no information about ground-truth communities and no consensus about an optimal community partition [35,36]. To generate sparse networks with missing links, we randomly remove a fraction r of links from the networks, and average the results for each value of r over 100 samplings.

Using these two networks, we compare the performance of the standard map equation, the Bayesian estimate of the map equation with different values of Dirichlet prior parameter a_α, and the degree-corrected stochastic block model [21,22]. We are interested in the number of communities, the partition similarities measured with the adjusted mutual information (AMI), and the predictive accuracy with cross-validation. Since the map equation and the degree-corrected stochastic block model use stochastic search algorithms to detect communities, we average the results over ten searches for each of the 100 network samplings.

We analyze the Bayesian approach for prior $a \sim \ln(V)$. For the node degree, therefore, we use $a_\alpha = C\ln(V)$, where $\alpha = 1\dots V$ and C is a constant that we need to specify. For the module degree, we use $a_{i\curvearrowright} = a_{i\curvearrowleft} = \nu_i C\ln(V)$, where $\nu_i = V_i\frac{V-V_i}{V-1}$ for $i = 1\dots M$ and V_i is the number of nodes in module i.

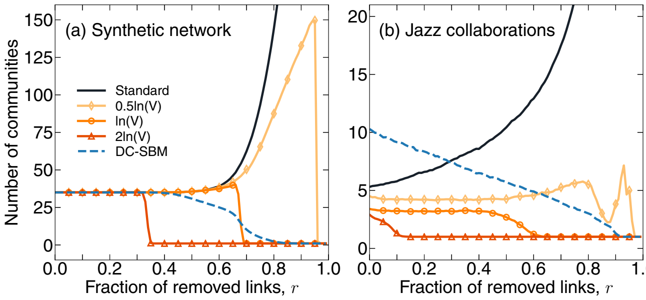

FIG. 3. Mean number of communities obtained by the standard map equation, the Bayesian estimate of the map equation with different values of Dirichlet prior parameter a, and the degree-corrected stochastic block model (DC-SBM). The Bayesian estimate of the map equation with prior $a_\alpha = \ln(V)$ provides the best solution: when sufficient network data are available it distinguishes significant communities from mere noise, while in the undersampled regime it detects no community structure. Results are averaged over 100 network samplings and ten algorithm searches. The standard error of the mean is never higher than 0.58.

A. Number of communities

Applied to the synthetic network, the standard map equation favors the planted partition until we remove more than approximately 55% of the links [Fig. 3(a)]. As we remove more links, the network also becomes sparse within communities. In the undersampled regime below the detectability limit where it is not possible to recover the planted partition, the map equation overfits to random fluctuations and favors more, smaller communities. The Bayesian estimate of the map equation behaves differently. For $C = 0.5$, the random prior network is weakly connected and cannot prevent overfitting when we remove 70–95% of the links. In contrast, for $C = 2$, the random prior network is densely connected and hides the communities in the noise induced by the prior such that the Bayesian estimate of the map equation underfits even when sufficient network data are available. In between, at the critical point where the random prior network becomes connected, the prior constant $C = 1$ balances over- and underfitting and prevents the detection of spurious communities. Moreover, the amount of noise that this prior network induces in the original network is so low that it does not wash out any significant community structure. While prior parameter C between 0.5 and 1 performs best for some analyzed networks, $C = 1$ remains a robust choice in general (Appendix C).

The degree-corrected stochastic block model detects the planted partition until we remove more than 40% of the links from the synthetic network. Compared to the Bayesian estimate of the map equation with the prior constant $C = 1$, the degree-corrected stochastic block model starts to underfit the planted partition earlier. For $r > 40\%$, the number of communities decreases continuously and when $r > 80\%$, the degree-corrected stochastic block model detects no community structure.

Similar behaviors appear accentuated when we apply the methods to the real-world Jazz collaboration network [Fig. 3(b)]. For the standard map equation, the number of detected communities increases with the number of missing links, whereas the degree-corrected stochastic block model shows the opposite trend. Unlike when applied to the synthetic network, the various map equation variants already favor different partitions before removing any links. The Bayesian estimate of the map equation detects fewer communities than the standard map equation, and its performance depends on the choice of the prior. For $C = 0.5$, the average number of communities is relatively stable when more than 50% of the links remain. However, if we remove more than 50% of the links, the number of communities increases because the prior parameter is too low. As for the synthetic network, the prior parameter $C = 2$ is too high and causes underfit: the method detects no community structure when we remove more than 10% of the links. Again, $C = 1$ offers a good tradeoff. The number of communities is approximately constant as long as at least 50% of the links remain and then decreases to 1 when fewer than 40% of the links remain, where the method deduces that there no longer exists any significant community structure.

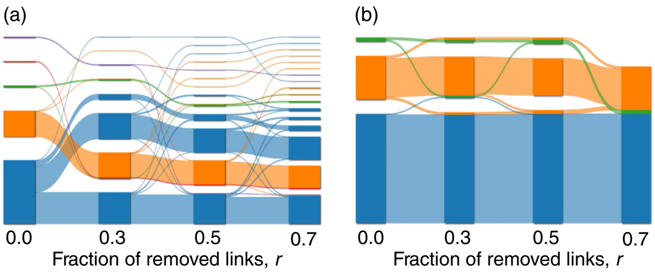

FIG. 4. Alluvial diagrams of the Jazz collaboration network show changes in community structure with missing links for (a) the standard map equation and (b) the Bayesian estimate of the map equation with prior $a_\alpha = \ln(V)$. Compared to the standard map equation, the communities detected using the Bayesian estimate of the map equation are more robust to missing links.

We illustrate differences in the community structure of the Jazz collaboration network induced by missing links for the standard and Bayesian map equation with alluvial diagrams [37]. The standard map equation identifies more and smaller communities with sparser networks, whereas its Bayesian estimate keeps similar communities with few changes before collapsing into one community when only 30% of the links remain. The Bayesian estimate's prior assumption of missing links prevents the map equation from splitting communities when the networks lose links (Fig. 4).

B. Adjusted mutual information

AMI is a standard measure used to compare two different partitions [38]. For the synthetic network, we compare identified partitions with the planted partition. The standard map equation successfully recovers the planted partition when more than 60% of the links are available (AMI = 1). When we remove more links, the accuracy decreases [Fig. 5(a)]. The Bayesian estimate of the map equation with prior constant $C = 0.5$ has almost the same accuracy. If we use $C = 1$ instead, then the method performs slightly better when we remove 40–60% of the links. Again, when we remove more than 65% of the links, the Bayesian estimate of the map equation with prior constant $C = 1$ deduces that there no longer exists any significant community structure and AMI = 0.

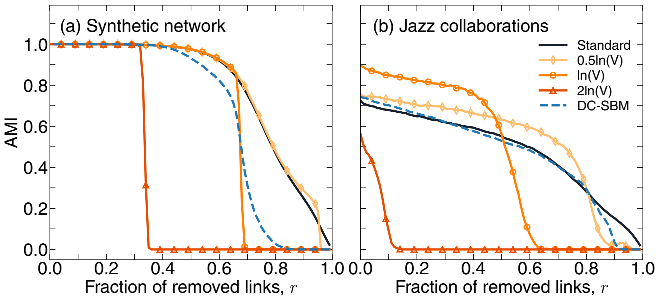

FIG. 5. Performance tests of the community-detection algorithms using AMI. (a) AMI scores with the planted partition of the synthetic network as reference. (b) AMI scores with a partition obtained for the complete Jazz collaboration network as reference. The Bayesian estimate of the map equation with prior $a_\alpha = \ln(V)$ gives the most robust results when it is possible to detect significant communities. Results are averaged over 100 network samplings and ten algorithm searches. The standard error of the mean is never higher than 0.01.

To measure the AMI for the Jazz collaboration network, which has no planted partition, we compare the partitions that the community detection methods return for networks with different fractions of missing links to the partitions they return for the complete network. For the complete network, we measure the average AMI over ten searches. The Bayesian estimate of the map equation with prior $a_\alpha = \ln(V)$ is the most consistent method when it is possible to detect significant communities [Fig. 5(b)].

In both synthetic and real-world networks when $\langle k\rangle > \ln(V)$, the Bayesian estimate of the map equation with prior constant $C = 1$ shows robust performance. However, when $C = 2$ it can fail to detect their community structure due to the high level of noise induced by the prior. To understand how the noise induced by the prior in the Bayesian estimate of the map equation affects community detection in sparse networks with $\langle k\rangle \sim \ln(V)$ and weak community structure, we test the performance on a range of different networks. We generate LFR networks with various values of average degree and mixing parameter, randomly remove a fraction of links, detect communities using the standard map equation and its Bayesian estimate with prior $a_\alpha = 0.5\ln(V)$ and $\ln(V)$, and classify the community detection as successful when the AMI between the planted partition and the identified partition is 0.9 or higher. Even if the random prior network has higher density than the original network, the Bayesian estimate of the map equation achieves the same performance as the standard map equation when the community structure is well defined ($\mu < 0.5$). However, if the community structure is weak ($\mu = 0.5$), then the prior $a_\alpha = \ln(V)$ can cause underfit before the standard map equation starts to overfit to noise induced by missing links (Fig. 6). These results rely on the cost of overfitting and underfitting implied by the AMI. Specific networks or research questions may require other penalties for many or few communities.

C. Cross-validation

Cross-validation allows us to compare model-selection performance without planted or known partitions. We validate the significance of network partitions returned by Infomap for training networks with a fraction $1 - r$ of links using the standard map equation and its Bayesian estimate (Fig. 7).

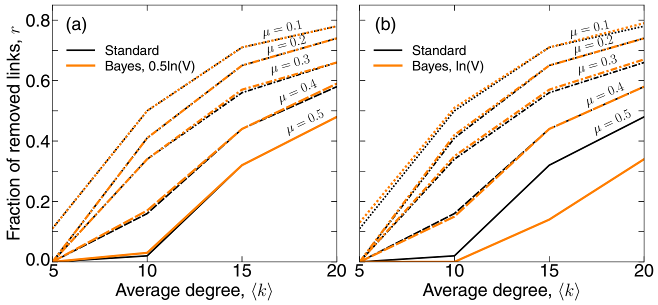

FIG. 6. Impact of network structure on the performance of the standard map equation and its Bayesian estimate. Prior parameter $C = 0.5$ in (a) and $C = 1$ in (b). For LFR networks with $V = 1000$ nodes and various densities $\langle k\rangle$ and mixing parameters μ, we show the critical fraction of removed links $r(\langle k\rangle, \mu)$ where the AMI between the planted partition and the identified partition falls below 0.9. Except for weak community structures ($\mu = 0.5$), where the Bayesian estimate with prior constant $C = 1$ underfits for lower fraction of removed links than the standard map equation overfits, the methods are on par. Results are averaged over ten network samplings and ten algorithm searches.

As the link density of the training network decreases below the detectability limit, the standard map equation mistakes noisy substructures in the sparse training networks for actual communities. As a result, the relative code length savings in the training and test networks diverge, and partitions obtained with the standard map equation give negative code length savings in the test network. In contrast, the Bayesian estimate of the map equation with prior constant $C = 1$ prevents overfitting in the sparse training network, implying that there is no significant community structure.

To complement with results for other networks, we provide summary statistics for six real-world networks often used to evaluate the performance of community detection algorithms (Table I). The networks include a collaboration network in Astrophysics extracted from the arXiv (AstroPh)

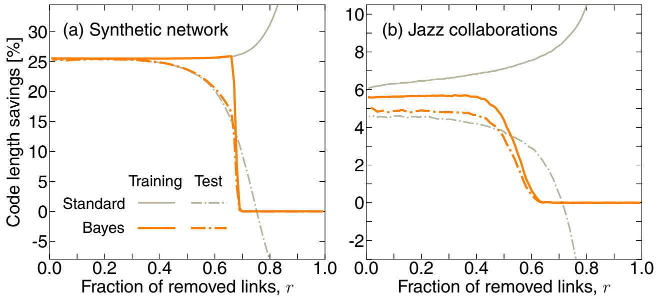

FIG. 7. Performance tests of the map equation with and without Bayesian estimates using cross-validation. The Bayesian estimate of the map equation with prior $a_\alpha = \ln(V)$ prevents overfitting in the undersampled regime. Results are averaged over 100 network samplings and ten algorithm searches. The code length is measured with Grassberger entropy estimation. The standard error of the mean is never higher than 0.38.

TABLE I. Comparison between partitions detected by the standard map equation and the Bayesian estimate of the map equation for six real-world networks. The notations $\mathrm{m}_{0.25}$ and $\mathrm{m}_{1.0}$ refer to the number of communities in the network with 25% removed links and the complete network, respectively. The last two columns report the code length savings of test and training networks for partitions detected in the training networks with 25% removed links

Network	Nodes	Links	Method	m_0	$\mathrm{m}_{0.25}$	$l_{0.25}^{\mathrm{train}}(\%)$	$l_{0.25}^{\mathrm{test}}(\%)$
AstroPh	17,903	197,031	Bayes	707	771	24	18
			Standard	663	1,080	24	18
Email	1,133	5,451	Bayes	34	1	0	0
			Standard	50	104	16	2
Erdős N1	466	1,600	Bayes	1	1	0	0
			Standard	38	67	17	−9
Football	115	613	Bayes	9	9	18	15
			Standard	10	11	20	16
PGP	10,680	24,316	Bayes	956	1,057	49	19
			Standard	897	2,210	49	16
Polblogs	1,222	16,717	Bayes	24	23	6	5
			Standard	33	80	6	5

[39], the network of e-mails exchanged between members of the University Rovira i Virgili (Email) [40], a collaboration network of authors with Erdős number 1 (Erdős N1) [41], the American College Football network (Football) [42], the PGP social network of trust (PGP) [43], and the network of political weblogs (Polblogs) [44]. In all networks, the standard map equation returns partitions with a higher number of communities when links are missing. Except for the Football network, the number of detected communities increases by 60% or more compared with the number of communities detected in the complete network. In contrast, except for the AstroPh and PGP networks, the Bayesian estimate of the map equation with prior constant $C = 1$ identifies partitions with fewer communities. Nevertheless, the different community structures detected by the two methods result in similar relative code length savings in all networks but the Email and Erdős N1 networks. They are sparse with $\langle k\rangle < \ln(V)$. In the complete Email network, the Bayesian estimate of the map equation detects 34 communities but underfits and detects no community structure after removing 25% of the links. After removing links in the Erdős N1 network, the standard map equation overfits and detects communities that, when applied to the test network, gives worse compression than the one-module solution. The Bayesian estimate of the map equation prevents this overfitting by preferring the one-module solution over any non-trivial solution.

Overall, the model-accuracy results quantified by number of communities, AMI scores, and code length savings in cross-validation on synthetic and real-world networks suggest that the analyzed network and research question should determine whether to use the standard map equation or its Bayesian estimate. Choose the standard map equation when the network data are complete or when extra communities caused by missing links are not a problem. Choose its Bayesian estimate when spurious communities can harm the analysis.

V. CONCLUSION

We have derived a Bayesian approach of the map equation that imposes prior information about the network structure to reduce overfitting for sparse networks with missing links. Using an uninformative Dirichlet prior, we show that the Bayesian estimate of the map equation avoids finding spurious communities in sparse synthetic and real-world networks with missing links. With a properly chosen prior constant, the proposed method successfully balances the impact of the imposed prior against the observed network data: The Bayesian estimate of the map equation provides a principled approach to reducing overfitting and detecting significant communities in two or more levels. We also show how to asses whether communities are significant using more effective cross-validation with Grassberger entropy estimation, which enables larger training networks. The computational overhead of the methods compared with the standard map equation is low. We anticipate that more reliable flow-based community detection of undersampled networks will be useful in many applications, including better prediction of missing links.

ACKNOWLEDGMENTS

We thank Vincenzo Nicosia and Leto Peel for stimulating discussions and Christopher Blöcker, Anton Eriksson, and Alexis Rojas for useful comments on the manuscript. J.S., D.E., and M.R. were supported by the Swedish Research Council, Grant No. 2016-00796.

APPENDIX A: NORMALIZED TRANSITION RATES

Proposition. The map equation,

$$\begin{aligned}L(\mathsf{M}) = &-\sum_{\alpha=1}^{V} p_\alpha \log_2(p_\alpha) - \sum_{i=1}^{m} q_{i\curvearrowright}\log_2(q_{i\curvearrowright})\\ &+\sum_{i=1}^{m}\left(q_{i\curvearrowright}+\sum_{\alpha\in i}p_\alpha\right)\log_2\left(q_{i\curvearrowright}+\sum_{\alpha\in i}p_\alpha\right)\\ &-\sum_{i=1}^{m} q_{i\curvearrowleft}\log_2(q_{i\curvearrowleft}) + \left(\sum_{i=1}^{m}q_{i\curvearrowleft}\right)\log_2\left(\sum_{i=1}^{m}q_{i\curvearrowleft}\right),\end{aligned} \tag{A1}$$

is a scale invariant function.

Proof. If we scale the transition rates p_α, $q_{i\curvearrowleft}$ and $q_{i\curvearrowright}$ by a constant K, where $K > 0$, and change $L(\mathsf{M})$ to

$$\begin{aligned}L'_{\mathsf{M}} = &-\sum_{\alpha=1}^{V} Kp_\alpha \log_2(Kp_\alpha) - \sum_{i=1}^{m} Kq_{i\curvearrowright}\log_2(Kq_{i\curvearrowright})\\ &+\sum_{i=1}^{m}\left(Kq_{i\curvearrowright}+\sum_{\alpha\in i}Kp_\alpha\right)\log_2\left(Kq_{i\curvearrowright}+\sum_{\alpha\in i}Kp_\alpha\right)\\ &-\sum_{i=1}^{m} Kq_{i\curvearrowleft}\log_2(Kq_{i\curvearrowleft})\\ &+\left(\sum_{i=1}^{m}Kq_{i\curvearrowleft}\right)\log_2\left(\sum_{i=1}^{m}Kq_{i\curvearrowleft}\right),\end{aligned}$$

then

$$\begin{aligned}
L'_{\mathsf{M}} = &-K\cancel{\sum_{\alpha=1}^{V} p_\alpha \log_2(K)} - K\sum_{\alpha=1}^{V} p_\alpha \log_2(p_\alpha)\\
&- K\cancel{\sum_{i=1}^{m} q_{i\curvearrowright} \log_2(K)} - K\sum_{i=1}^{m} q_{i\curvearrowright} \log_2(q_{i\curvearrowright})\\
&- K\cancel{\sum_{i=1}^{m} q_{i\curvearrowleft} \log_2(K)} - K\sum_{i=1}^{m} q_{i\curvearrowleft} \log_2(q_{i\curvearrowleft})\\
&+ K\cancel{\sum_{i=1}^{m} q_{i\curvearrowright} \log_2(K)} + K\cancel{\sum_{\alpha=1}^{V} p_\alpha \log_2(K)}\\
&+ K\sum_{i=1}^{m}\left(q_{i\curvearrowright} + \sum_{\alpha\in i} p_\alpha\right)\log_2\left(q_{i\curvearrowright} + \sum_{\alpha\in i} p_\alpha\right)\\
&+ K\cancel{\left(\sum_{i=1}^{m} q_{i\curvearrowleft}\right)\log_2(K)}\\
&+ K\left(\sum_{i=1}^{m} q_{i\curvearrowleft}\right)\log_2\left(\sum_{i=1}^{m} q_{i\curvearrowleft}\right)\\
= &\, KL(\mathsf{M}).
\end{aligned}$$

■

If we choose

$$K = \frac{\sum_{\alpha=1}^{V} k_\alpha}{\sum_{\alpha=1}^{V} k_\alpha + \sum_{i=1}^{m} k_{i\curvearrowleft} + \sum_{i=1}^{m} k_{i\curvearrowright}}, \tag{A2}$$

such that

$$p'_\alpha = Kp_\alpha = \frac{k_\alpha}{\sum_{\alpha=1}^{V} k_\alpha + \sum_{i=1}^{m} k_{i\curvearrowleft} + \sum_{i=1}^{m} k_{i\curvearrowright}}, \tag{A3}$$

$$q'_{i\curvearrowleft} = Kq_{i\curvearrowleft} = \frac{k_{i\curvearrowleft}}{\sum_{\alpha=1}^{V} k_\alpha + \sum_{i=1}^{m} k_{i\curvearrowleft} + \sum_{i=1}^{m} k_{i\curvearrowright}}, \tag{A4}$$

$$q'_{i\curvearrowright} = Kq_{i\curvearrowright} = \frac{k_{i\curvearrowright}}{\sum_{\alpha=1}^{V} k_\alpha + \sum_{i=1}^{m} k_{i\curvearrowleft} + \sum_{i=1}^{m} k_{i\curvearrowright}}, \tag{A5}$$

then we will have

$$\sum_{\alpha=1}^{V} p'_\alpha + \sum_{i=1}^{m} q'_{i\curvearrowleft} + \sum_{i=1}^{m} q'_{i\curvearrowright} = 1. \tag{A6}$$

Now we can use

$$\begin{aligned}
L(\mathsf{M}) = \frac{1}{K}\Bigg[&-\sum_{\alpha=1}^{V} p'_\alpha \log_2(p'_\alpha) - \sum_{i=1}^{m} q'_{i\curvearrowright}\log_2(q'_{i\curvearrowright})\\
&+ \sum_{i=1}^{m}\left(q'_{i\curvearrowright} + \sum_{\alpha\in i} p'_\alpha\right)\log_2\left(q'_{i\curvearrowright} + \sum_{\alpha\in i} p'_\alpha\right)\\
&- \sum_{i=1}^{m} q'_{i\curvearrowleft}\log_2(q'_{i\curvearrowleft}) + \left(\sum_{i=1}^{m} q'_{i\curvearrowleft}\right)\log_2\left(\sum_{i=1}^{m} q'_{i\curvearrowleft}\right)\Bigg]
\end{aligned} \tag{A7}$$

to calculate the posterior average of the map equation

$$\begin{aligned}
\hat{L}_B(\mathsf{M}) &= E[L(\mathsf{M})|k, a]\\
&= \int L(\mathsf{M})P(p', q'_{\curvearrowleft}, q'_{\curvearrowright}|k, a)\,dp'\,dq'_{\curvearrowleft}\,dq'_{\curvearrowright},
\end{aligned} \tag{A8}$$

where posterior probability distribution equals

$$\begin{aligned}
&P(\boldsymbol{p}', \boldsymbol{q}'_{\curvearrowleft}, \boldsymbol{q}'_{\curvearrowright}|\boldsymbol{k}, \boldsymbol{a})\\
&\quad\propto \prod_{\alpha=1}^{V}(p'_\alpha)^{k_\alpha + a_\alpha - 1}\prod_{i=1}^{m}[(q'_{i\curvearrowright})^{k_{i\curvearrowright}+a_{i\curvearrowright}-1}(q'_{i\curvearrowleft})^{k_{i\curvearrowleft}+a_{i\curvearrowleft}-1}].
\end{aligned} \tag{A9}$$

As a result, we obtain

$$\begin{aligned}
\hat{L}_B(\mathsf{M}) = &\frac{1}{\ln(2)}\frac{1}{\sum_{\alpha=1}^{V} u_\alpha}\\
&\times\Bigg[-\sum_{\alpha=1}^{V} u_\alpha\psi(u_\alpha + 1) - \sum_{i=1}^{m} u_{i\curvearrowright}\psi(u_{i\curvearrowright} + 1)\\
&+ \sum_{i=1}^{m}\left(u_{i\curvearrowright} + \sum_{\alpha\in i} u_\alpha\right)\psi\left(u_{i\curvearrowright} + \sum_{\alpha\in i} u_\alpha + 1\right)\\
&- \sum_{i=1}^{m} u_{i\curvearrowleft}\psi(u_{i\curvearrowleft} + 1)\\
&+ \left(\sum_{i=1}^{m} u_{i\curvearrowleft}\right)\psi\left(\sum_{i=1}^{m} u_{i\curvearrowleft} + 1\right)\Bigg],
\end{aligned} \tag{A10}$$

where $u_x = k_x + a_x$ and ψ is digamma function, $\psi(x) = \frac{d}{dx}\ln[\Gamma(x)]$.

APPENDIX B: THE BAYESIAN ESTIMATE OF THE MULTILEVEL MAP EQUATION

The multilevel formulation of the map equation [6,31] measures the minimum average description length given a multilevel map M of V nodes clustered into m communities, for which each community i has a submap M_i with m_i subcommunities, for which each subcommunity ij has a submap M_{ij} with m_{ij} subcommunities, and so on. It uses hierarchically nested code structures,

$$L(\mathsf{M}) = q_{\curvearrowleft}H(\mathcal{Q}) + \sum_{i=1}^{m} L(\mathsf{M}_i), \tag{B1}$$

where the average per-step code length needed to describe random walker transitions between communities at the coarsest level is the same as in the case of two-level clusterings,

$$H(\mathcal{Q}) = -\sum_{i=1}^{m}\frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}}\log_2\frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}}, \tag{B2}$$

and the average per-step code word length of the module codebook i recursively takes into account contributions of the description lengths of communities at finer levels,

$$L(\mathsf{M}_i) = q_i^{\circlearrowright}H(\mathcal{Q}_i) + \sum_{i=1}^{m_i} L(\mathsf{M}_{ij}). \tag{B3}$$

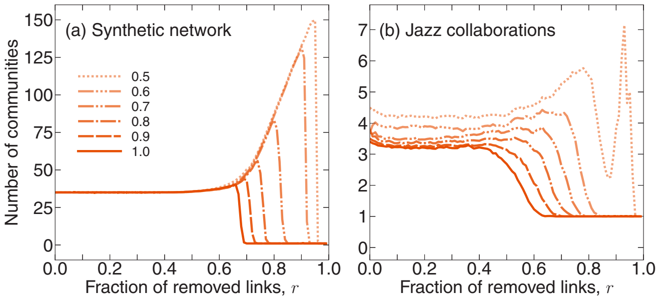

FIG. 8. Mean number of communities obtained by the Bayesian estimate of the map equation with different values of the prior constant C. Smaller prior constants give more communities when many links are missing. Results are averaged over 100 network samplings and ten algorithm searches.

Here, the average per-step code length needed to describe the random walker at intermediate level i exiting to a coarser level or entering the m_i subcommunities M_{ij} at a finer level is

$$H(\mathcal{Q}_i) = -\frac{q_{i\curvearrowright}}{q_i^{\circlearrowright}}\log_2\frac{q_{i\curvearrowright}}{q_i^{\circlearrowright}} - \sum_{j=1}^{m_i}\frac{q_{ij\curvearrowleft}}{q_i^{\circlearrowright}}\log_2\frac{q_{ij\curvearrowleft}}{q_i^{\circlearrowright}}, \qquad \text{(B4)}$$

where

$$q_i^{\circlearrowright} = q_{i\curvearrowright} + \sum_{j=1}^{m_i} q_{ij\curvearrowleft} \qquad \text{(B5)}$$

is the total code rate use in subcommunity i. We add the description lengths of codebooks for subcommunities at finer levels in a recursive fashion down to the finest level,

$$L(\mathsf{M}_{ij\ldots l}) = p_{ij\ldots l}^{\circlearrowright} H(\mathcal{P}_{ij\ldots l}), \qquad \text{(B6)}$$

where

$$H(\mathcal{P}_{ij\ldots l}) = -\frac{q_{ij\ldots l\curvearrowright}}{p_{ij\ldots l}^{\circlearrowright}}\log_2\frac{q_{ij\ldots l\curvearrowright}}{p_{ij\ldots l}^{\circlearrowright}} - \sum_{\alpha\in\mathsf{M}_{ij\ldots l}}\frac{\pi_\alpha}{p_{ij\ldots l}^{\circlearrowright}}\log_2\frac{\pi_\alpha}{p_{ij\ldots l}^{\circlearrowright}} \qquad \text{(B7)}$$

and

$$p_{ij\ldots l}^{\circlearrowright} = q_{ij\ldots l\curvearrowright} + \sum_{\alpha\in\mathsf{M}_{ij\ldots l}}\pi_\alpha \qquad \text{(B8)}$$

is the total code word use rate of module codebook $ij\ldots l$.

To obtain the Bayesian estimate of the multilevel map equation, we use Eq. (B1) to calculate the posterior average according to Eq. (4). Following the same procedure described in Sec. III A, we obtain a formula for the Bayesian estimate of the multilevel map equation,

$$\hat{L}_B(\mathsf{M}) = \frac{1}{\ln(2)}\frac{1}{\sum_{\alpha=1}^{V}u_\alpha}\left[-\sum_{i=1}^{m}u_{i\curvearrowleft}\psi(u_{i\curvearrowleft}+1) + \left(\sum_{i=1}^{m}u_{i\curvearrowleft}\right)\psi\left(\sum_{i=1}^{m}u_{i\curvearrowleft}+1\right)\right] + \sum_{i=1}^{m}\hat{L}_B(\mathsf{M}_i), \qquad \text{(B9)}$$

where

$$\hat{L}_B(\mathsf{M}_i) = \frac{1}{\ln(2)}\frac{1}{\sum_{\alpha=1}^{V}u_\alpha}\left[-u_{i\curvearrowright}\psi(u_{i\curvearrowright}+1) - \sum_{j=1}^{m_i}u_{ij\curvearrowleft}\psi(u_{ij\curvearrowleft}+1) + \left(u_{i\curvearrowright}+\sum_{j=1}^{m_i}u_{ij\curvearrowleft}\right)\psi\left(u_{i\curvearrowright}+\sum_{j=1}^{m_i}u_{ij\curvearrowleft}+1\right)\right] + \sum_{j=1}^{m_i}\hat{L}_B(\mathsf{M}_{ij}) \qquad \text{(B10)}$$

and at the finest level

$$\hat{L}_B(\mathsf{M}_{ij\ldots l}) = \frac{1}{\ln(2)}\frac{1}{\sum_{\alpha=1}^{V}u_\alpha}\left[-u_{ij\ldots l\curvearrowright}\psi(u_{ij\ldots l\curvearrowright}+1) - \sum_{\alpha\in\mathsf{M}_{ij\ldots l}}u_\alpha\psi(u_\alpha+1) + \left(u_{ij\ldots l\curvearrowright}+\sum_{\alpha\in\mathsf{M}_{ij\ldots l}}u_\alpha\right)\psi\left(u_{ij\ldots l\curvearrowright}+\sum_{\alpha\in\mathsf{M}_{ij\ldots l}}u_\alpha+1\right)\right]. \qquad \text{(B11)}$$

APPENDIX C: RESULTS FOR DIFFERENT VALUES OF THE PRIOR PARAMETER

The number of communities obtained by the Bayesian estimate of the map equation varies for different values of the prior constant C between 0.5 and 1 (Fig. 8). For the synthetic network in the undersampled regime, $C < 0.8$ can lead to severe overfitting before removing so many links that it becomes evident that there is no significant community structure. For the Jazz collaboration network, the number of detected communities is similar for prior constant $C > 0.6$ but is higher for all values of r when $C \leqslant 0.6$.

To compare the performance for different prior parameters, we also compute the AMI for C between 0.5 and 1 (Fig. 9). For the synthetic network, the AMI results confirm that the detected communities become sensitive to the choice of prior when we remove more than 65% of the links. For example, for $C \geqslant 0.8$, the detected communities have AMI down to 0.65 before dropping to 0. For $C < 0.8$, the method can detect communities in sparser networks but these communities have

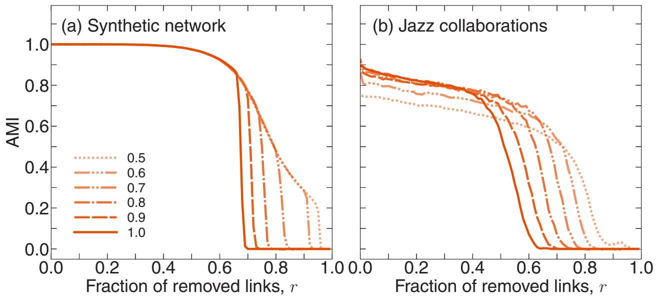

FIG. 9. Performance tests of the Bayesian estimate of the map equation with different values of the prior constant C using AMI. (a) AMI scores with the planted partition of the synthetic network as reference. (b) AMI scores with a partition obtained for the complete Jazz collaboration network as reference. Smaller prior constants give communities with non-zero AMI scores when many links are missing at the cost of overall lower AMI-scores in the Jazz network. Results are averaged over 100 network samplings and ten algorithm searches.

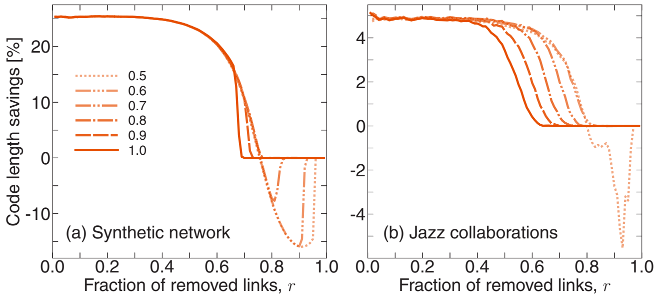

FIG. 10. Performance tests of the Bayesian estimate of the map equation with different values of the prior constant C using cross-validation. Smaller prior constants give higher compression in a narrow range of missing links at the cost of lower compression for more missing links. We show relative code length savings for the test network compared to the one-community partition. The code length is measured with Grassberger entropy estimation. Results are averaged over 100 network samplings and ten algorithm searches.

AMI scores below 0.5. For the Jazz collaboration network, the AMI results confirm that the detected communities are more robust when $C > 0.6$.

Cross-validation further confirms these results for different prior parameters. For the synthetic network, the Bayesian estimate of the map equation is more robust to overfitting with prior constant $C \geqslant 0.8$ (Fig. 10). With $C < 0.8$ and more than 75% of the links removed, the communities detected in the training network applied to the test network give worse compression than with a single community. For the Jazz collaboration network, a prior with $C \geqslant 0.6$ prevents the detection of communities in the training network that, when applied to the test network, give negative relative code length savings.

These results for different values of the prior parameter indicate that there is no single prior $C \ln(V)$ that achieves optimal performance for all networks. We suggest using $\ln(V)$ as a prior because it is robust to overfitting and has good overall performance. If desired for specific networks, then C can be optimized between 0.5 and 1 with cross-validation.

[1] P. Pons and M. Latapy, J. Graph Algor. Appl. **10**, 191 (2006).
[2] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).
[3] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, Proc. Natl. Acad. Sci. USA **107**, 12755 (2010).
[4] M. T. Schaub, J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, PLoS One **7**, e32210 (2012).
[5] R. Lambiotte, J. Delvenne, and M. Barahona, IEEE Trans. Network Sci. Eng. **1**, 76 (2014).
[6] D. Edler, L. Bohlin, and M. Rosvall, Algorithms **10**, 112 (2017).
[7] R. Guimerà and M. Sales-Pardo, Proc. Natl. Acad. Sci. USA **106**, 22073 (2009).
[8] L. Lu and T. Zhou, Physica A: Stat. Mech. Appl. **390**, 1150 (2011).
[9] A. Ghasemian, H. Hosseinmardi, and A. Clauset, IEEE Trans. Knowl. Data Eng. **1**, 1 (2019).
[10] T. Martin, B. Ball, and M. E. J. Newman, Phys. Rev. E **93**, 012306 (2016).
[11] M. E. J. Newman, Nat. Phys. **14**, 542 (2018).
[12] M. E. J. Newman, Phys. Rev. E **98**, 062321 (2018).
[13] T. P. Peixoto, Phys. Rev. X **8**, 041011 (2018).
[14] T. Squartini, G. Caldarelli, G. Cimini, A. Gabrielli, and D. Garlaschelli, Phys. Rep. **757**, 1 (2018).
[15] C. E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948).
[16] G. P. Basharin, Theory Probab. Appl. **4**, 333 (1959).
[17] T. P. Peixoto, Phys. Rev. X **4**, 011047 (2014).
[18] T. Vallès-Català, T. P. Peixoto, M. Sales-Pardo, and R. Guimerà, Phys. Rev. E **97**, 062316 (2018).
[19] D. H. Wolpert and D. R. Wolf, Phys. Rev. E **52**, 6841 (1995).
[20] P. Grassberger, arXiv:physics/0307138.
[21] T. P. Peixoto, Phys. Rev. E **95**, 012317 (2017).
[22] T. P. Peixoto, arXiv:2003.07070.
[23] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 056117 (2009).
[24] G. Miller, in *Information Theory in Psychology; Problems and Methods*, edited by H. Quastler (Free Press, Glencoe, IL, 1955).
[25] S. Zahl, Ecology **58**, 907 (1977).
[26] I. Nemenman, F. Shafee, and W. Bialek, in *Advances in Neural Information Processing Systems 14* (MIT Press, Cambridge, MA, 2002).
[27] E. Archer, I. M. Park, and J. W. Pillow, J. Mach. Learn. Res. **15**, 2833 (2014).
[28] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis* (Cambridge University Press, New York, NY, 2005).
[29] P. Erdős and A. Rényi, Publ. Math. Debrecen **6**, 290 (1959).
[30] D. Edler, A. Eriksson, and M. Rosvall, The Infomap Software Package (2020), https://www.mapequation.org.
[31] M. Rosvall and C. T. Bergstrom, PLoS One **6**, e18209 (2011).
[32] T. Schürmann, J. Phys. A **37**, L295 (2004).
[33] P. M. Gleiser and L. Danon, Adv. Complex Syst. **06**, 565 (2003).

[34] A. Lancichinetti, S. Fortunato, and F. Radicchi, Phys. Rev. E **78**, 046110 (2008).

[35] M. E. J. Newman, Phys. Rev. E **94**, 052315 (2016).

[36] L. Peel, D. B. Larremore, and A. Clauset, Sci. Adv. **3**, e1602548 (2017).

[37] M. Rosvall and C. T. Bergstrom, PLoS One **5**, e8694 (2010).

[38] N. X. Vinh, J. Epps, and J. Bailey, J. Mach. Learn. Res. **11**, 2837 (2010).

[39] J. Leskovec, J. Kleinberg, and C. Faloutsos, ACM Trans. Knowl. Discovery Data **1**, 2 (2007).

[40] H. Ebel, L.-I. Mielsch, and S. Bornholdt, Phys. Rev. E **66**, 035103(R) (2002).

[41] V. Batagelj and A. Mrvar, Soc. Networks **22**, 173 (2000).

[42] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).

[43] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, Phys. Rev. E **70**, 056122 (2004).

[44] L. A. Adamic and N. Glance, in *Proceedings of the Workshop on the Weblogging Ecosystem (WWW'05)* (ACM, New York, 2005).

Journal of Complex Networks (2021) **6**, Advance Access Publication on 6 December 2021
doi: 10.1093/comnet/cnab044

Mapping flows on weighted and directed networks with incomplete observations

Jelena Smiljanić[†]
Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden and Scientific Computing Laboratory, Center for the Study of Complex Systems, Institute of Physics Belgrade, University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia
[†]Corresponding author. Email: jelena.smiljanic@umu.se

Christopher Blöcker
Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden

Daniel Edler
Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden, Gothenburg Global Biodiversity Centre, Box 461, SE-405 30 Gothenburg, Sweden and Department of Biological and Environmental Sciences, University of Gothenburg, Carl Skottsbergs Gata 22B, Gothenburg 41319, Sweden

and

Martin Rosvall
Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden

Edited by: Petter Holme

[Received on 19 July 2021; editorial decision on 28 October 2021; accepted on 28 October 2021]

Detecting significant community structure in networks with incomplete observations is challenging because the evidence for specific solutions fades away with missing data. For example, recent research shows that flow-based community detection methods can highlight spurious communities in sparse undirected and unweighted networks with missing links. Current Bayesian approaches developed to overcome this problem do not work for incomplete observations in weighted and directed networks that describe network flows. To overcome this gap, we extend the idea behind the Bayesian estimate of the map equation for unweighted and undirected networks to enable more robust community detection in weighted and directed networks. We derive an empirical Bayes estimate of the transitions rates that can incorporate metadata information and show how an efficient implementation in the community-detection method Infomap provides more reliable communities even with a significant fraction of data missing.

Keywords: community detection, directed and weighted networks, incomplete data, the map equation

1. Introduction

Network models gain explainable power with additional information about node labels or link directions and weights [1, 2]. But these data can also introduce uncertainties such as mislabelled nodes or noisy link measurements that the network methods must address for reliable further analysis [3]. For example, when community-detection methods disregard uncertainties in network data, they can overfit and generate inaccurate node classifications that affect downstream analyses such as link prediction [4–6].

To assess the significance of detected communities, we can statistically compare them with expected results under a null model [7, 8] or test how robust they are under random perturbations of the network [9]. However, both approaches are computationally expensive and impractical for large networks. Instead, we can integrate regularization mechanisms in the community-detection methods themselves to prevent them from capitalizing on spurious communities. Several community detection methods take this approach for undirected and unweighted networks. For example, community-detection methods based on statistical inference can incorporate assumptions about unreliable measurements into the generative network models [10, 11]. For the flow-based community-detection method known as the map equation, which identifies modular structure by searching for sets of nodes with long flow persistence [12, 13], we have derived a Bayesian estimate that copes with missing unweighted and undirected links [6]. However, dealing with incomplete observations for robust flow-based community detection in directed and weighted networks remains unresolved.

Since link weights and directions naturally describe network flows, the map equation works effectively for directed and weighted networks. But the Bayesian estimate of the map equation for unweighted and undirected links requires an analytical expression for the network-flow distribution. For directed networks, no such analytical solution exists. Because the Bayesian estimate of the map equation also assumes a binary network to derive link probabilities, it cannot be applied directly to weighted and directed networks.

Instead, we start from the basic idea behind the Bayesian estimate of the map equation and derive an empirical Bayes estimate of the transition rates between nodes in weighted, directed networks. Our Bayesian estimate employs the continuous configuration model [14] and gives a teleportation-like dynamics in a principled way with critical improvements for robust community detection. To ensure an ergodic stationary flow distribution in directed networks, standard teleportation turns a random walker into a random surfer that, besides following links proportional to their weights, teleports uniformly to nodes—connected or disconnected—at a fixed rate. However, teleporting at a fixed rate disregards basic network structure and can wash out significant communities, underfitting the data [15]. Other approaches that reduce the teleportation rate's influence on the community assignments can instead lead to overfitting in networks with missing data. In our Bayesian estimate of the transition rates, the network flows depend on the amount of available data and network type for robust flow-based community detection in unipartite or bipartite weighted, directed networks with or without metadata (Fig. 1).

We provide an implementation in Infomap that runs at native speed, available for anyone to download from https://www.mapequation.org. Using synthetic networks with planted community structures and real-world networks with varying fraction of link observations, we evaluate the empirical Bayes estimate of the transition rates. We find that Infomap with and without regularized network flows detects similar and robust communities when enough observations are available. But for incomplete networks with many missing observations, Infomap with empirical Bayes estimates of the transition rates outperforms standard Infomap and prevents spurious communities.

2. Methodology

The map equation is an information-theoretic objective function for detecting flow-based communities [12, 13]. Conceptually, it models network flows as random walks, encodes random walker movements between nodes using codewords, and estimates the theoretical lower limit of the average per-step codelength for a given partition of the nodes into modules. In line with the minimum description length principle, finding the partition that best compresses the network flows is equivalent to identifying most modular regularities in the network data with respect to those flows. The Infomap software package [16]

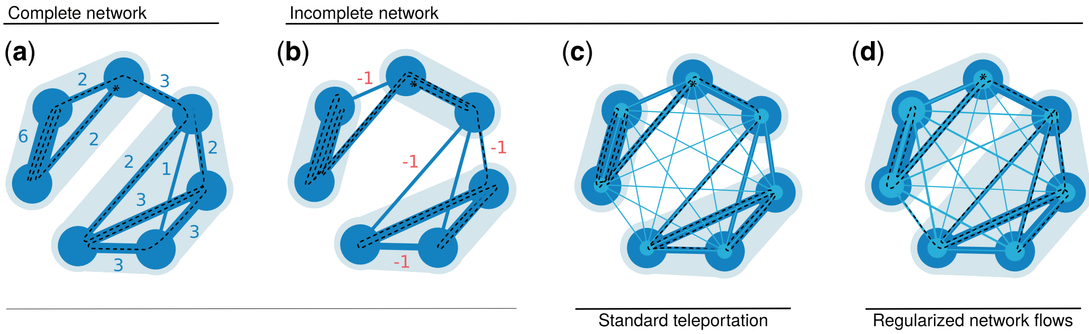

FIG. 1. A schematic weighted network with complete and missing link observations. (a) A complete network with accurate network flows and inferred communities. (b) Missing link observations introduce inaccuracies. (c) A standard teleportation scheme cannot overcome the inaccuracies. (d) Regularized network flows with an empirical Bayes estimate of the transition rates using the relaxed continuous configuration model recovers the complete network's community structure. Light background areas indicate optimal community assignments. The width of the light blue lines represents teleportation weight. The size of the light blue node centres indicates teleportation probability. The dashed black lines show sample trajectories of random walks. We omit link directions in this example for simplicity.

implements a fast and greedy search algorithm that maximizes flow compression over node partitions by minimizing the map equation.

The basic idea behind the map equation is a communication game where a sender uses codewords to update a receiver about the location of the random walker in the network. In a one-level partition without modular structure, we assign unique codewords to nodes, and the sender communicates one codeword per random-walker step to the receiver. The lower limit for the codelength is the Shannon entropy over the nodes' stationary visit rates according to Shannon's source coding theorem [17]. When partitioning nodes into more than one module, we can re-use codewords across modules and achieve shorter average codelengths. We introduce an index-level codebook to encode transitions between modules and one exit codeword per module for a uniquely decodable code. The sender uses one codeword to describe transitions within modules and three codewords between modules: for exiting the old module, for entering the new module, and for communicating the visited node in the new module. In the same fashion, we can extend the coding scheme to hierarchies with three or more levels. The partition that compresses the flows on the network the most reflects the network's community structure regarding that flow the best.

When sufficiently many observations are available, Infomap returns reliable communities [18, 19]. Because the map equation describes the network as-is, missing observations can misrepresent the actual stationary flow distribution, change the balance between module- and index-level codebooks, and distort the communities. As a consequence, the map equation may capitalize on noise and detect spurious partitions with more and smaller communities than actually present in the complete network [4, 6].

A Bayesian estimate that incorporates prior network assumptions into the map equation overcomes this overfitting problem, and can be derived in closed form for unweighted undirected networks where the stationary visit rate for node i is determined by its degree, k_i, as $p_i = \frac{k_i}{\sum_{i=1}^{N} k_i}$ [6, 20]. However, we cannot directly apply this approach to directed or weighted networks for two reasons. First, we cannot express a corresponding Bayesian estimate of the map equation analytically because no closed-form solution exists for node visit rates in directed networks. Second, the prior for weighted networks must incorporate link weights absent in previous work [6]. Instead, we formulate an empirical Bayes estimate of a random walker's transition rates to regularize node visit rates [21].

2.1 *The map equation with a Bayesian estimate of the transition rates*

We consider a weighted directed network with N nodes where A represents the adjacency matrix and the matrix W contains information on observed link weights. We assume integer weights for simplicity, but the method also works for non-negative real weights. In general, the probabilities that a random walker steps from node i to other nodes are given by $T_i = (t_{i1}, \ldots, t_{iN})$. If we interpret the network as a multigraph, such that w_{ij} denotes the number of observed links between nodes i and j, we can explain $W_i = (w_{i1}, \ldots, w_{iN})$ as a sample of the hidden distribution T_i. Estimating transition rates t_{ij} using the maximum likelihood estimator gives

$$\tilde{t}_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}. \tag{2.1}$$

However, with noisy data, $\tilde{t}_{ij}$ can deviate significantly from t_{ij} and cause the map equation to overfit the observed data. To prevent the map equation from overfitting and increase its generalizability, we regularize the transition rates using a Bayesian approach [21]. We introduce a prior distribution over T_i and estimate posterior transition rates

$$\hat{t}_{ij}(W_i) = \int t_{ij} P(T_i|W_i) dT_i, \tag{2.2}$$

where $P(T_i|W_i)$ is a posterior over the unknown distribution T_i given by Bayes' rule,

$$P(T_i|W_i) = \frac{P(W_i|T_i)P(T_i)}{P(W_i)}. \tag{2.3}$$

As prior distribution $P(T_i)$, we choose the Dirichlet distribution, which is the conjugate prior of the multinomial distribution and enables analytical calculations:

$$P(T_i|\gamma_i) = \frac{\Gamma(\gamma_{i1} + \cdots + \gamma_{iN})}{\Gamma(\gamma_{i1}) \ldots \Gamma(\gamma_{iN})} \prod_{j=1}^{N} t_{ij}^{\gamma_{ij}-1}. \tag{2.4}$$

$\Gamma(x)$ is the gamma function and $\gamma_{i1} \ldots \gamma_{iN}$ are parameters of the distribution. Given that the likelihood

$$P(W_i|T_i) = (w_{i1} + \cdots + w_{iN})! \prod_{j=1}^{N} \frac{t_{ij}^{w_{ij}}}{w_{ij}!} \tag{2.5}$$

and the total probability of the data

$$P(W_i) = \int P(W_i|T_i)P(T_i) dT_i, \tag{2.6}$$

the posterior distribution

$$P(T_i|W_i,\gamma_i) \propto \prod_{j=1}^{N} t_{ij}^{w_{ij}+\gamma_{ij}-1}. \tag{2.7}$$

Finally, after integrating Eq. 2.2, we obtain

$$\hat{t}_{ij} = \frac{w_{ij}+\gamma_{ij}}{\sum_{j=1}^{N} w_{ij}+\gamma_{ij}} \tag{2.8}$$

$$= (1-\alpha_i)\frac{w_{ij}}{\sum_j w_{ij}} + \alpha_i \frac{\gamma_{ij}}{\sum_j \gamma_{ij}}, \tag{2.9}$$

where $\alpha_i = \frac{\sum_{j=1}^{N}\gamma_{ij}}{\sum_{j=1}^{N} w_{ij}+\gamma_{ij}}$. The first term is the maximum likelihood estimator weighted by $(1-\alpha_i)$ and the second term is the transition rates from the prior distribution weighted by α_i. Together they form our empirical Bayes estimate of the transition rates.

The effect of this Bayesian estimate on the transition rates resemble modelling network flows with teleportation. Standard teleportation allows a random walker to teleport uniformly to any node in the network with a fixed small probability α independent of the visited node i. Teleportation is necessary to ensure ergodicity in directed networks [22] but disregards the network structure and turns the flow distribution dependent on the teleportation parameter α [15]. For the problem of missing observations, teleportation is not a viable option: For low teleportation rates, the network structure dominates such that the map equation can overfit to noise in the data (Fig. 1(c)). Conversely, for high teleportation rates, random jumps dominate over the network structure such that the map equation can underfit and fail to detect relevant community structures.

Interpreting the Bayesian estimate of the transition rates in terms of teleportation, Eq. (2.9) shows that a random walker has node-dependent source and target teleportation probabilities. The random walker chooses an observed link with probability $1-\alpha_i$, or a link in the fully connected prior network with probability α_i. In both cases, the probability to follow a link (i,j) is proportional to its observed weight w_{ij} and prior weight γ_{ij}, respectively. Thus, if node i has many out-links, the random walker will likely follow them. Otherwise, if the number of out-links of node i is small, it will teleport with a higher probability (Fig. 1(d)).

How the method performs depends on the parameters γ. We should choose them such that they can reduce bias induced by incomplete observations while still not wash out regularities in the network structure. We assume that the adjacency matrix A and the weight matrix W are decoupled and use

$$\gamma_{ij} = \lambda_{ij} c_{ij}, \tag{2.10}$$

where λ_{ij} is a connectivity parameter that reflects our prior assumption about connections between nodes i and j and the weight parameter c_{ij} reflects our belief about link weights.

2.2 *The connectivity parameter*

We use the connectivity parameter $\lambda_{ij} = \lambda = \frac{\ln N}{N}$, which corresponds to the connectivity threshold of random networks. This λ-value is the theoretical lower bound on density that guarantees almost surely a giant connected component in the network [23, 24]. When no further node attributes are known, we assume that the connectivity between each pair of nodes is $\lambda = \frac{\ln N}{N}$. This choice creates a prior network strong enough to prevent overfitting but permissive enough to detect well-supported communities, and works well to regularize the map equation for undirected, unweighted networks [6]. The choice manifests a prior belief that the network is connected without any community structure. When more information about nodes is available, such as types, classes, or similar, the connectivity parameter, λ_{ij}, should be adjusted to reflect this information. We consider two concrete cases, bipartite networks and nodes annotated with metadata.

2.2.1 *Bipartite networks.* Bipartite networks model interactions between two kinds of node types, A and B, where only nodes with different types interact directly. A connectivity of $\lambda = \frac{\ln N}{N}$ between all pairs of nodes violates the bipartite structure of the network. To preserve the bipartite nature of the network, we set the connectivity parameter for links between same-type nodes to zero and adjust it for links between different-type nodes.

We assume a bipartite network with N_A nodes of type A, N_B nodes of type B, and uniform distribution of links between different-type nodes. As before, we pick the smallest connectivity parameter λ_{AB} such that the resulting network is almost surely connected, $\lambda_{AB} = \frac{\ln\left(N_A+N_B\right)}{\min\left(N_A,N_B\right)}$ [25]. The resulting bipartite prior weight between nodes i and j, using bipartite connectivity λ_{AB}, is

$$\gamma_{ij}^{\text{bi}} = \left(1 - \delta_{t_i t_j}\right) \lambda_{AB} c_{ij}, \tag{2.11}$$

where t_i and t_j are the types of nodes i and j, respectively, and δ is the Kronecker delta.

2.2.2 *Metadata.* Real-world networks often contain more information than links. For example, nodes can have additional metadata. Metadata have primarily aided in interpreting detected communities. However, recent studies suggest that complementing network data with metadata for community detection can help overcome limitations and uncertainties in the network structure [26–29].

We use discrete metadata to adjust the connectivity parameter. As before, we connect each pair of nodes uniformly with connectivity $\lambda = \frac{\ln N}{N}$. In addition, we use the metadata and reinforce connections between nodes with the same label m by $\lambda_m = \frac{\ln N_m}{N_m}$, where N_m is the number of nodes with label m. With metadata labels m_i and m_j for nodes i and j, respectively, the adjusted prior link weight is

$$\gamma_{ij}^{\text{meta}} = \left(\lambda + \delta_{m_i m_j} \lambda_{m_i}\right) c_{ij}. \tag{2.12}$$

2.3 *Weight parameter*

To incorporate prior assumptions on weights into our method, we use an empirical Bayesian approach [30]. An uninformative prior, such as an exponential link weight distribution, is inadequate since it can wash out regularities in the network structure. Instead, we assume that the data carry information about their prior distribution and estimate prior link weights from the networks.

To derive link weights for a prior network, we adapt the so-called continuous configuration model [14], which estimates the weight of the link from node i to j as

$$c_{ij} = \frac{\sum_{n=1}^{N} k_n^{\text{in}} + k_n^{\text{out}}}{\sum_{n=1}^{N} s_n^{\text{in}} + s_n^{\text{out}}} \frac{s_i^{\text{out}} s_j^{\text{in}}}{k_i^{\text{out}} k_j^{\text{in}}}, \tag{2.13}$$

where k_i^{in} and k_i^{out} denote observed in- and out-degrees, and $s_i^{\text{in}} = \sum_j w_{ji}$ and $s_i^{\text{out}} = \sum_j w_{ij}$ denote in- and out-strengths for node i. The connectivity parameters defined by Eq. (2.13) preserve expected weights of in- and out- links incident to a node. They provide higher link weights between nodes with strong connections to their neighbours.

This method also works for unweighted and undirected networks. Undirected networks can be considered as a special case of directed networks where $k_i^{\text{out}} = k_i^{\text{in}} = k_i$ and $s_i^{\text{out}} = s_i^{\text{in}} = s_i$ for all nodes i. The relaxed continuous configuration model assigns weights $c_{ij} = 1$ to all links for unweighted networks. In this case, our method presented here and the Bayesian estimate of the map equation [6] provide identical results. While we can express the effect of the prior network analytically in the Bayesian estimate of the map equation for undirected, unweighted networks, we can also express it as a Bayesian estimate of the transition rates as in Eq. (2.9) and use it with the standard map equation.

We provide an efficient implementation of the Bayesian estimate of the transition rates for anyone to download from https://www.mapequation.org. The general implementation for regularized network flows works for unipartite and bipartite, unweighted and weighted, undirected and directed networks with and without metadata. The code runs at native speed because it does not express the all-to-all transition rates from the prior distribution in Eq. (2.9) as links.

3. Results

We evaluate the performance of Infomap with our empirical Bayes estimate of the transition rates in networks with missing observations. Our focus is on weighted, directed networks with unweighted and undirected networks as special cases. For simplicity, we restrict our analyses to networks with integer weights and interpret them as multigraphs, such that link weights w_{ij} denote the number of observed edges between nodes i and j. To create networks with missing observations, we sample from synthetic and empirical multigraphs by removing an r-fraction of their multiedges uniformly at random, resulting in reduced edge weights. For robust results, we average over 100 repetitions for each r-value. As a baseline, we use the performance of the standard map equation and compare the number of detected communities, partition similarity and predictive accuracy. We measure partition similarity with the adjusted mutual information (AMI) [31] between detected and planted partition and predictive accuracy with cross-validation.

3.1 *Synthetic networks*

We use the Lancichinetti–Fortunato–Radicchi (LFR) method [18] to generate a weighted directed network with $N = 1000$ nodes, average node degree $k = 7$, and mixing parameter $\eta = 0.4$. The resulting network has $M = 31$ communities and an average link weight of 4.9 with integer link weights. We have included results for synthetic networks with different parameters in Appendix A.

To construct synthetic networks with metadata, we first assign metadata labels in perfect alignment with the community assignments of the LFR networks. Because metadata labels and network community structure are not always aligned [32], we assign one of the existing $M = 31$ metadata labels to a μ-fraction of the nodes at random to evaluate the performance for different metadata and community structure correlations. In this way, we can use the same network to test our empirical Bayes estimate of the transition rates both with and without metadata.

With uniform connectivity and as long as we remove up to half of the edges, corresponding to $r \leq 0.5$, the standard map equation and the map equation with regularized network flows detect virtually the same number of communities [Fig. 2(a)]. When we remove more than half of the data and move beyond $r = 0.5$,

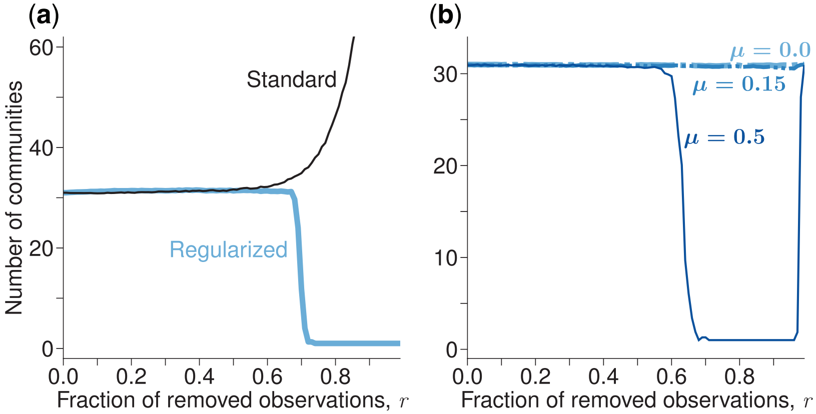

FIG. 2. Mean number of communities in synthetic weighted and directed networks with and without our empirical Bayes estimate of transition rates. Without metadata in (a) and with metadata in (b), where a fraction μ of the nodes have randomly assigned metadata. Results are averages over 100 network samplings.

the standard map equation begins to detect more and smaller communities. In contrast, the map equation with regularized network flows does not detect community structure anymore. The relative weight of the prior network increases as we remove more data and the remaining evidence is not strong enough to support communities.

With a metadata-based Bayesian estimate of the transition rates, the fraction of removed links, r, does not affect the number of detected communities if the correlation between metadata and planted partition, μ, is high [Fig. 2(b)]. When we randomize half of the metadata labels, corresponding to $\mu = 0.5$, and move beyond the detectability point at $r \approx 0.65$, we find two regimes. First, two opposing forces are at work, the noisy network structure and the metadata, and we detect no community structure. Then, as we approach $r = 1$ and almost no link observations remain in the network, we detect the partition corresponding to the metadata labels.

Although the standard map equation detects the correct number of $M = 31$ communities when we remove less than half of the observations, the AMI scores show that Infomap assigns some nodes to incorrect communities [Fig. 3(a)]. The map equation with regularized network flows detects communities that better match the planted communities. When we remove more than half of the observations, $r > 0.5$, the standard map equation detects more communities and the AMI score decreases. In contrast, the map equation with regularized network flows detects only one community with an AMI score of zero, indicating that the available data is insufficient to infer community structure.

When using a metadata-based Bayesian estimate of the transition rates, our method detects the planted partition reliably if the metadata and the planted partition match perfectly, corresponding to $\mu = 0$ [Fig. 3(b)]. The method assigns some nodes incorrectly for $\mu > 0$ and weaker correlations with less aligned structural and metadata information. When many observations are missing, the performance depends on how well the metadata align with the planted community structure.

Many communities and low AMI scores in the undersampled regime indicate that the standard map equation returns spurious communities. To understand better how this affects the system characterization, we use a cross-validation approach where we first split the multiedge counts of a network into training and test multiedges such that the same edge (i, j) can occur in the training and validation data, and their counts sum to the original observed count. Then, we infer the partition that maximizes compression in

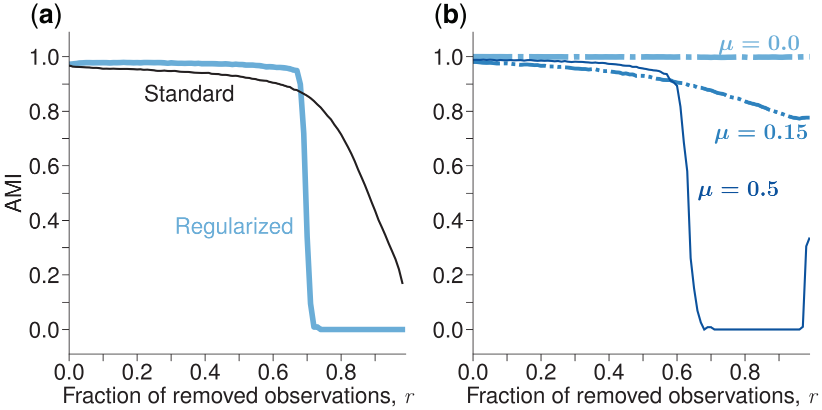

FIG. 3. Adjusted mutual information in synthetic weighted and directed networks with and without Bayesian estimate of the transition rates. Without metadata in (a) and with metadata in (b), where a fraction μ of the nodes have randomly assigned metadata. Results are averages over 100 network samplings.

the training network with Infomap and calculate the test network's description length using that partition. If the partition captures the structure of the training network well, we expect that it also compresses the description length in the test network. However, if insufficient data are available in the training network, Infomap overfits and returns a partition that inaccurately describes the structure of the test network, resulting in low compression. Since the modular description length depends on the number of link observations [6], we construct balanced two-fold splits. For a multigraph with m observed edges, we choose $\frac{m}{2}$ edges uniformly at random and without replacement for the training network and place the remaining $\frac{m}{2}$ edges in the test network. Because this split induces further undersampling, we cannot compare the link-removal performance with the previous analysis that started with a complete network. Nevertheless, we can use the results to provide more insights into how each method performs in the undersampled regime.

To quantify the level of compression that a partition M achieves in the test network, we consider the relative codelength savings, the codelength for partition M compared to the one-module solution M_1 that assigns all nodes to the same module, $l = 1 - \frac{L(\mathsf{M})}{L(\mathsf{M}_1)}$. Although the standard map equation does not find the optimal partition under incomplete observations, the results indicate that it captures some regularities and achieves positive codelength savings [Fig. 4(a)]. However, when the codelength savings are negative, a correct delineation of the network structure is likely infeasible. The map equation with regularized network flows and uniform connectivity achieves better compression up until $r \approx 0.4$, indicating that it better captures the network structure. Beyond this point, and in the regime where the standard map equation detects partitions with negative compression, the map equation with regularized network flows without metadata information assigns all nodes to the same community, resulting in no compression and codelength savings of zero [Fig. 4(a)].

The map equation with metadata-based Bayesian estimate of the transition rates detects partitions that capture the network regularities well and provide positive codelength savings, even when the metadata labels do not match the planted community assignments for a moderate fraction of the nodes, for example, $\mu = 0.15$. [Fig. 4(b)]. However, when the correlation between metadata and planted partition is weak ($\mu = 0.5$), and many observations are missing, the method cannot identify significant communities anymore.

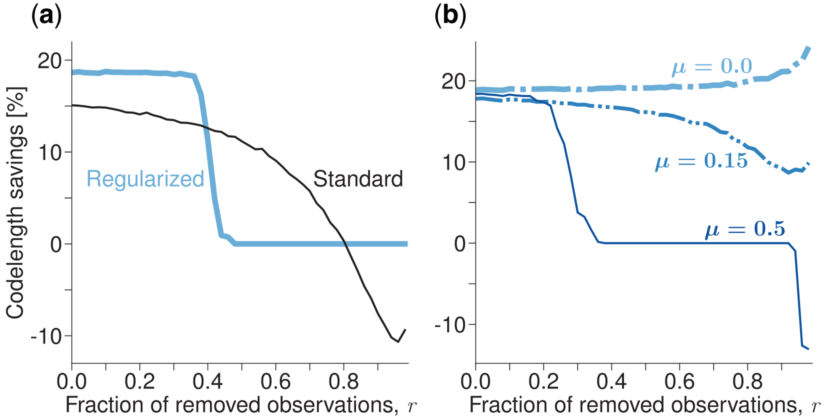

FIG. 4. Codelength savings in synthetic weighted and directed networks with and without regularized network flows. Without metadata in (a) and with metadata in (b), where a fraction μ of the nodes have randomly assigned metadata. Results are averages over 100 network samplings.

3.2 *Empirical networks*

We analyse the performance of the map equation with and without regularized network flows on six empirical networks from different domains where four of the networks are weighted, and three are directed.

Sociopatterns The social network of recorded interactions between female and male students in a high school in Marseille organized as bipartite network [33]. The students are assigned to one of nine classes which we use as metadata.

CoRA The network covers citations between computer science research papers [34]. The papers are classified into nine different research topics that we use as metadata.

Industry The network contains companies that are connected if they appeared together in a business story [34]. We use Yahoo!'s 12 industry sectors as metadata.

cit-HepTh The network contains citations from within arXiv's HEP-TH section [35]. We consider only published articles and use information about the journals as metadata.

Pokémon Using information from all seven generations of Pokémon, we create a network by connecting two Pokémon who share the same abilities [36]. The primary type of the Pokémon is used as metadata.

Openflights The network contains links between non-USA airports [37]. We use countries as metadata.

Table 1 provides summary information of topological properties of the networks and their metadata.

We analyse each of the six empirical networks and report the number of communities (Fig. 5) and relative codelength savings (Fig. 6). However, because there is no ground truth partition for empirical data, we cannot use AMI to evaluate our results.

TABLE 1 *Summary of network data. The column Kind denotes if the network is directed (D) or undirected (U). The notations w and M refer to the average link weight and the number of metadata categories in the network, respectively. The last column reports the AMI between metadata and partition detected by the standard map equation in the complete network*

Network	Nodes	Links	Kind	w	M	AMI
Sociopatterns [33]	143+175	2265	U	1.33	9	0.9
CoRA [34]	3385	22092	D	1.00	9	0.3
Industry [34]	1778	14154	U	2.79	12	0.2
cit-HepTh [35]	4378	55186	D	1.00	9	0.0
Pokémon [36]	743	18184	U	1.10	18	0.3
Opeflights [37]	964	8850	D	1.48	97	0.4

The empirical networks behave like the synthetic networks when analysed with the standard map equation and the map equation with regularized network flows. In the complete networks, and when we remove only a small fraction of the observations, the methods detect partitions with a similar number of communities. When we remove more observations and enter the undersampled regime, the standard map equation detects more and smaller communities. In contrast, the map equation with regularized network flows without metadata information detects no community structure (Fig. 5).

The empirical networks enter the undersampled regime at different points. In the Pokémon and Industry networks, the map equation with regularized network flows detects communities even after removing 70% of the observations. In the Pokémon network, the number of communities detected by the map equation with regularized network flows increases slightly with the fraction of removed observations before it drops sharply to 1 at $r = 0.8$ and no community structure is detected anymore. However, the community structure in the cit-HepTh network is sensitive to undersampling, and the map equation with regularized network flows cannot detect communities if we remove more than 5% of the observations.

The cross-validation results show that partitions with noisy substructures detected by the standard map equation sometimes compress flows on the test network better than the one-level partition. With more data missing, eventually, the detected partitions lead to negative codelength savings, and the one-level partition offers a better description of the network flows (Fig. 6). The map equation with teleportation does not suffer from this issue. The mechanism we have implemented prevents overfitting and instead returns the one-level partition when not enough data is available to support community structure in the network.

How well metadata labels align with the network structure determines the performance for the map equation with regularized network flows using metadata. We use the partitions detected by the standard map equation on the complete networks as a proxy for the network structures and report the AMI with the metadata labels in Table 1. For example, in the Sociopatterns network, the metadata contains useful information and improves the performance in the undersampled regime. The number of communities remains the same for all r-values [Fig. 5(a)] and, as the cross-validation results show, we achieve high compression in the test network [Fig. 6(a)]. In contrast, in the cit-HepTh network, where journals do not support citation patterns between articles, the metadata does not reveal significant communities in the network structure [Fig. 5(d)]. Similarly, in the Pokémon network where metadata labels align only weakly with community structure, we observe lower performance than for the map equation with regularized network flows without employing metadata. When we remove almost all link observations, using uncorrelated metadata can lead to negative codelength savings [Fig. 6(d and e)].

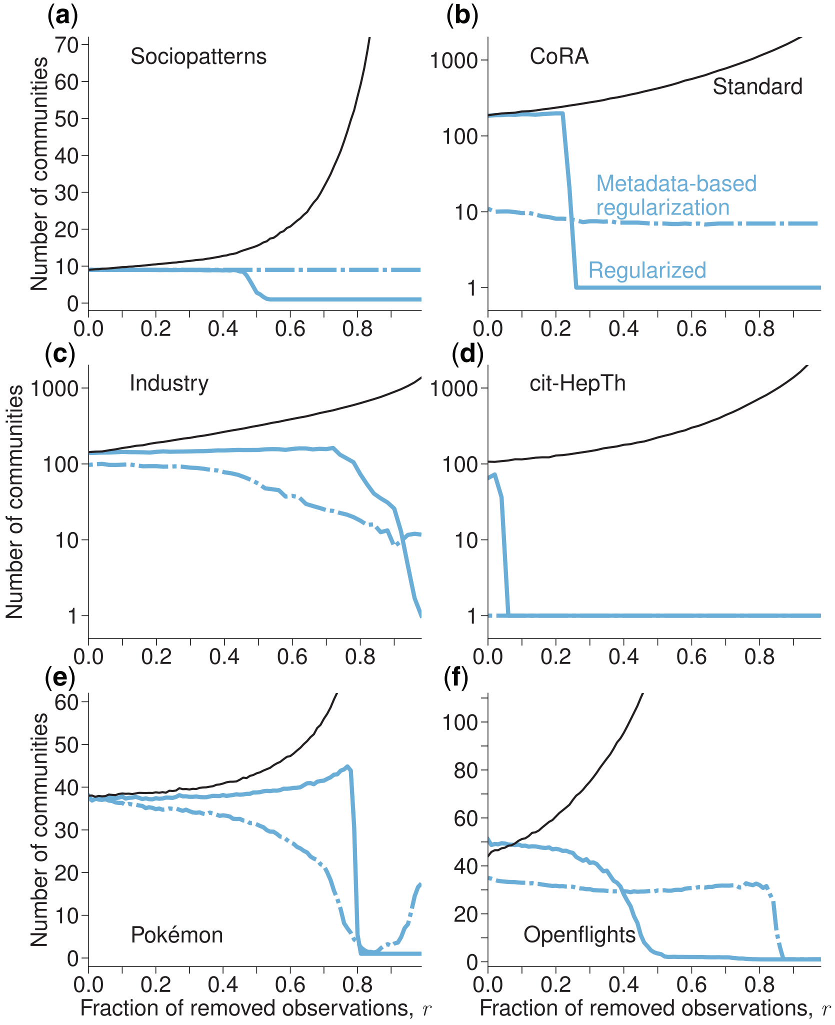

FIG. 5. Mean number of communities in empirical networks obtained by the standard map equation, the map equation with teleportation and uniform connectivity, and the map equation with metadata-based Bayesian estimate of the transition rates. Results are averages over 100 network samplings.

In the remaining three networks, even though the correlation between metadata and community structure is low, we find that the map equation with regularized network flows benefits from employing the metadata in the undersampled regime. The map equation with metadata-based Bayesian estimate of the transition rates detects fewer communities than the other two methods. The higher codelength savings indicate that the detected partitions better capture the structural patterns in the networks by avoiding overfitting to weakly supported substructures [Fig. 6(b, c and f)].

Our analyses show that using regularized network flows with or without metadata prevents overfitting in the undersampled regime. Instead of returning spurious partitions from sparse observations, the map

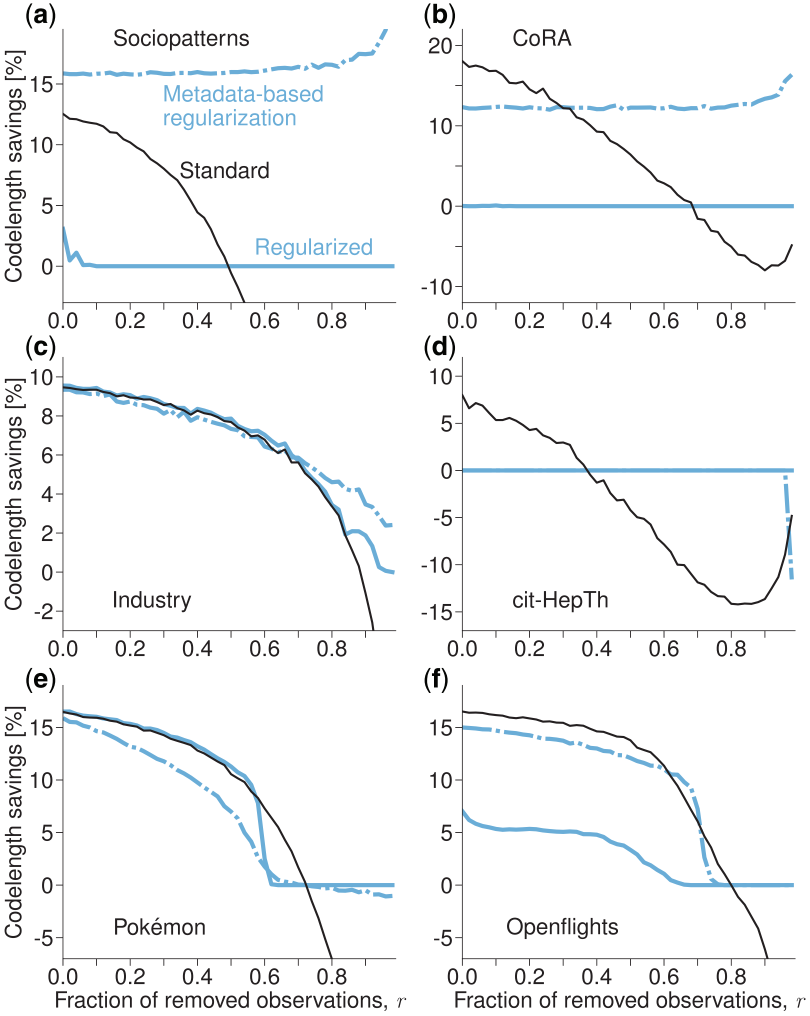

FIG. 6. Codelength savings in test networks obtained by the standard map equation, the map equation with teleportation and uniform connectivity, and the map equation with metadata-based Bayesian estimate of the transition rates. Results are averages over 100 network samplings.

equation with regularized network flows returns the one-level partition, indicating insufficient evidence to support any community structure. To detect more regularities with better compression, the metadata-based Bayesian estimate of the transition rates detects more regularities and achieves better compression when correlations between metadata and the network structure are moderate or higher. With low correlations, the map equation with regularized network flows with metadata can underfit, and the map equation with regularized network flows without employing metadata performs better.

Overall, we recommend the standard map equation for complete network data or when communities from missing links are not problematic. When spurious communities can harm the analysis, the map equation with regularized network flows provides a robust approach.

4. Conclusion

We have equipped the flow-based map equation framework with a regulatory mechanism to deal with missing link observations in weighted and directed networks. By deriving an empirical Bayes estimate of the transition rates that employs a relaxed continuous configuration model, the network flow dynamics account for the uncertainty of observed node degrees and strengths. The empirical Bayes estimate of the transition rates can incorporate additional information about node types and attributes, enabling extensions to bipartite networks and networks with metadata. Our adaptable solution also supersedes artificial teleportation for mathematically sound flow modelling on directed networks.

We have implemented the map equation with empirical Bayes estimates of the transition rates in Infomap and analysed synthetic and real-world networks to evaluate the performance. Our results show that regularizing the network flows prevents overfitting in undersampled networks, even when a substantial fraction of the data are missing. Incorporating metadata to reflect prior knowledge about the network can compensate for missing link observations when the metadata correlate with the network structure. Our results suggest that the map equation with an empirical Bayes estimate of the transition rates provides an effective way to identify robust communities in weighted and directed networks with incomplete observations.

Code

We have implemented the map equation with our Bayesian estimate of the transition rates in Infomap. Full documentation of Infomap, including tutorials, instructions and visualization tools is available at https://www.mapequation.org.

Funding

The Wallenberg AI, Autonomous Systems and Software Program (https://wasp-sweden.org) funded by the Knut and Alice Wallenberg Foundation to C.B.; the Swedish Research Council (2016-00796) to D.E., J.S. and M.R.

References

1. Barrat, A., Barthelemy, M., Pastor-Satorras, R. & Vespignani, A. (2004) The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. USA*, **101**, 3747–3752.
2. Newman, M. E. J. (2004) Analysis of weighted networks. *Phys. Rev. E*, **70**, 056131.
3. Newman, M. E. J. (2018) Network structure from rich but noisy data. *Nat. Phys.*, **14**, 542–545.
4. Ghasemian, A., Hosseinmardi, H. & Clauset, A. (2019) Evaluating overfit and underfit in models of network community structure. *IEEE Trans. Knowl. Data Eng.*, **32**, 1722–1735.
5. Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M. & Clauset, A. (2020) Stacking models for nearly optimal link prediction in complex networks. *Proc. Natl. Acad. Sci. USA*, **117**, 23393–23400.
6. Smiljanić, J., Edler, D. & Rosvall, M. (2020) Mapping flows on sparse networks with missing links. *Phys. Rev. E*, **102**, 012302.
7. Lancichinetti, A., Radicchi, F. & Ramasco, J. J. (2010) Statistical significance of communities in networks. *Phys. Rev. E*, **81**, 046110.
8. Lancichinetti, A., Radicchi, F., Ramasco, J. J. & Fortunato, S. (2011) Finding statistically significant communities in networks. *PLoS One*, **6**, 1–18.
9. Rosvall, M. & Bergstrom, C. T. (2010) Mapping change in large networks. *PLoS One*, **5**, 1–7.
10. Martin, T., Ball, B. & Newman, M. E. J. (2016) Structural inference for uncertain networks. *Phys. Rev. E*, **93**, 012306.

11. Peixoto, T. P. (2018) Reconstructing networks with unknown and heterogeneous errors. *Phys. Rev. X*, **8**, 041011.

12. Rosvall, M. & Bergstrom, C. T. (2008) Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA*, **105**, 1118–1123.

13. Edler, D., Bohlin, L. & Rosvall, M. (2017) Mapping higher-order network flows in memory and multilayer networks with Infomap. *Algorithms*, **10**, 112.

14. Palowitch, J., Bhamidi, S. & Nobel, A. B. (2018) Significance-based community detection in weighted networks. *J. Mach. Learn. Res.*, **18**, 1–48.

15. Lambiotte, R. & Rosvall, M. (2012) Ranking and clustering of nodes in networks with smart teleportation. *Phys. Rev. E*, **85**, 056107.

16. Edler, D., Eriksson, A. & Rosvall, M. (2020) *The Infomap Software Package*.

17. Shannon, C. E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.*, **27**, 379–423.

18. Lancichinetti, A. & Fortunato, S. (2009) Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, **80**, 016118.

19. Hric, D., Darst, R. K. & Fortunato, S. (2014) Community detection in networks: Structural communities versus ground truth. *Phys. Rev. E*, **90**, 062805.

20. Mitzenmacher, M. & Upfal, E. (2005) *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. New York, NY: Cambridge University Press.

21. Wang, X., Tao, T., Sun, J.-T., Shakery, A. & Zhai, C. (2008) DirichletRank: solving the zero-one gap problem of PageRank. *ACM Trans. Inf. Syst.*, **26**, 1–29.

22. Brin, S. & Page, L. (1998) The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN*, **30**, 107–117.

23. Erdős, P. & Rényi, A. (1959) On Random Graphs. *Publ. Math. Debrecen*, **6**, 290–297.

24. Palásti, I. (1966) On the strong connectedness of directed random graphs. *Studia Sci. Math. Hungar*, **1**, 205–214.

25. Saltykov, A. I. (1995) The number of components in a random bipartite graph. *Discrete Math. Appl.*, **5**, 515–524.

26. Yang, J., McAuley, J. & Leskovec, J. (2013) Community detection in networks with node attributes. *2013 IEEE 13th International Conference on Data Mining*. pp. 1151–1156.

27. Newman, M. E. J. & Clauset, A. (2015) Structure and inference in annotated networks. *Nat. Commun.*, **7**, 11863.

28. Hric, D., Peixoto, T. P. & Fortunato, S. (2016) Network structure, metadata, and the prediction of missing nodes and annotations. *Phys. Rev. X*, **6**, 031038.

29. Emmons, S. & Mucha, P. J. (2019) Map equation with metadata: varying the role of attributes in community detection. *Phys. Rev. E*, **100**, 022301.

30. Efron, B. (2010) *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Vol. 1, Institute of Mathematical Statistics Monographs, Cambridge University Press.

31. Vinh, N. X., Epps, J. & Bailey, J. (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, **11**, 2837–2854.

32. Peel, L., Larremore, D. B. & Clauset, A. (2017) The ground truth about metadata and community detection in networks. *Sci. Adv.*, **3**, e1602548.

33. Mastrandrea, R., Fournet, J. & Barrat, A. (2015) Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS One*, **10**, 1–26.

34. Macskassy, S. A. & Provost, F. (2007) Classification in networked data: a toolkit and a univariate case study. *J. Mach. Learn. Res.*, **8**, 935–983.

35. Leskovec, J., Kleinberg, J. & Faloutsos, C. (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining.* ACM, pp. 177–187.

36. Banik, R. (2018) *The Complete Pokemon Dataset*.

37. Opsahl, T. (2011) Why anchorage is not (that) important: binary ties and sample selection. https://toreopsahl.com/2011/08/12/why-anchorage-is-not-that-important-binary-ties-and-sample-selection/.

A. Results for different configurations of synthetic networks

To understand how the Bayesian estimate of the transition rates affects community detection in networks with different structures, we test the performance on synthetic networks with various sizes, densities, and community strengths. We create six weighted directed LFR networks with various number of nodes, N, average degree, k and mixing parameter, η, then randomly remove an r-fraction of the link observations and detect communities with the standard map equation and the map equation with regularized network flows.

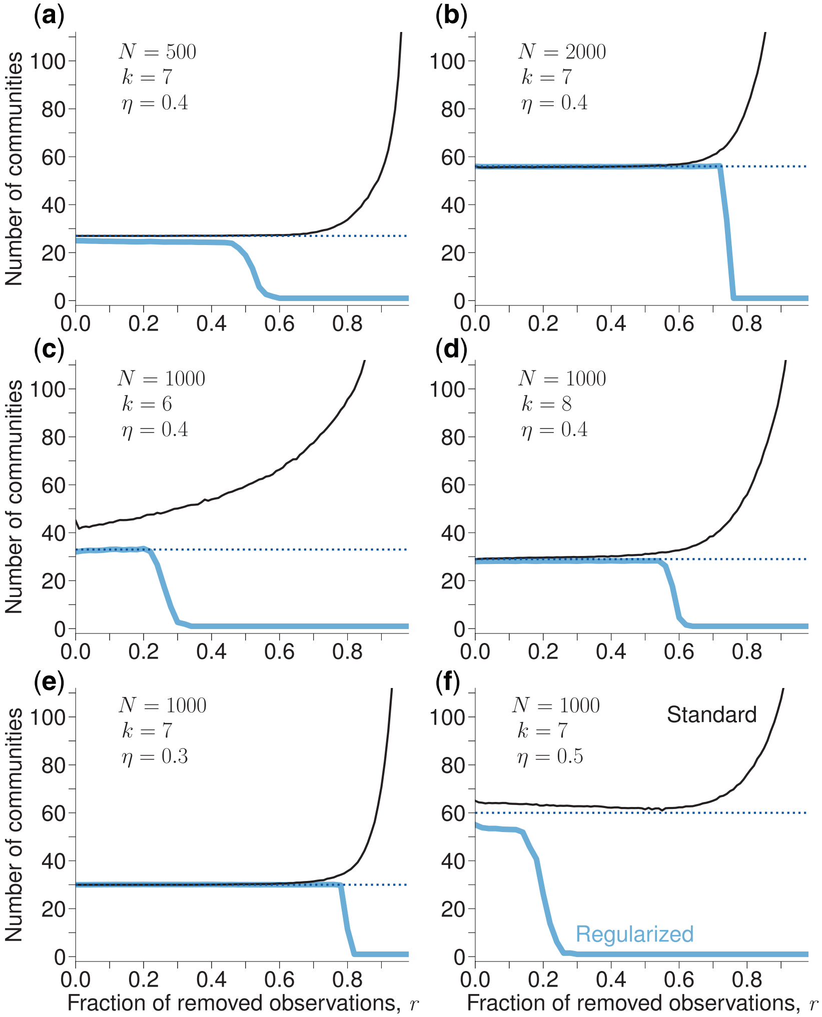

FIG. A.1. Mean number of communities in synthetic weighted and directed networks with and without regularized network flows. Dotted line indicates number of planted communities.

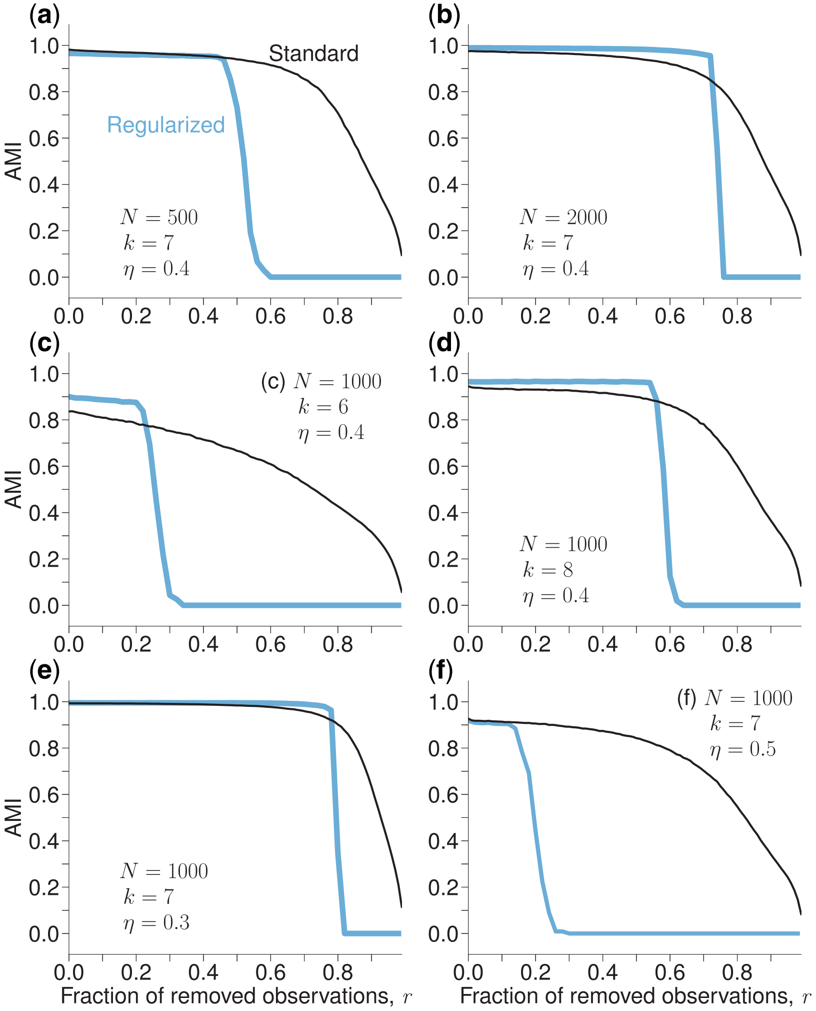

FIG. A.2. Adjusted mutual information in synthetic weighted and directed networks with and without regularized network flows.

Our results show similar trends in terms of robustness to noise in all six networks (Figs A.1 and A.2). In the undersampled regime, the performance of the standard map equation decreases fast as the number of missing observations increases. The map equation with regularized network flows undergoes a sharp transition from detecting robust communities to not detecting any community structure. The uninformative assumption that a network has no modular structure prevents the map equation with regularized network flows from detecting modular regularities in networks with weak community structure [Fig. A.2(f)]. However, in sparse networks with stronger support for community structure, we find that our Bayesian estimate of the transition rates can improve detection accuracy significantly [Fig. A.2(c)].

Journal of Statistical Mechanics:
Theory and Experiment

PAPER

Universal growth of social groups: empirical analysis and modeling

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: Interdisciplinary statistical mechanics

Universal growth of social groups: empirical analysis and modeling

Ana Vranić[1,*], Jelena Smiljanić[1,2] and Marija Mitrović Dankulov[1]

[1] Institute of Physics Belgrade, University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia
[2] Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden
E-mail: ana.vranic@ipb.ac.rs, jelena.smiljanic@ipb.ac.rs and marija.mitrovic.dankulov@ipb.ac.rs

Received 15 June 2022
Accepted for publication 28 October 2022
Published 7 December 2022

Online at stacks.iop.org/JSTAT/2022/123402
https://doi.org/10.1088/1742-5468/aca0e9

Abstract. Social groups are fundamental elements of any social system. Their emergence and evolution are closely related to the structure and dynamics of a social system. Research on social groups was primarily focused on the growth and the structure of the interaction networks of social system members and how members' group affiliation influences the evolution of these networks. The distribution of groups' size and how members join groups has not been investigated in detail. Here we combine statistical physics and complex network theory tools to analyze the distribution of group sizes in three data sets, Meetup groups based in London and New York and Reddit. We show that all three distributions exhibit log-normal behavior that indicates universal growth patterns in these systems. We propose a theoretical model that combines social and random diffusion of members between groups to simulate the roles of social interactions and members' interest in the growth of social groups. The simulation results show that our model reproduces growth patterns observed in empirical data. Moreover, our analysis shows that social interactions are more critical for the diffusion of members in online groups, such as Reddit, than in offline groups, such as Meetup. This work shows that social groups follow universal growth mechanisms that need to be considered in modeling the evolution of social systems.

[*] Author to whom any correspondence should be addressed.

Keywords: network dynamics, random graphs, networks, scaling in socio-economic systems, stochastic processes

Contents

1. Introduction

The need to develop methods and tools for their analysis and modeling comes with massive data sets. Methods and paradigms from statistical physics have proven to be very useful in studying the structure and dynamics of social systems [1]. The main argument for using statistical physics to study social systems is that they consist of many interacting elements. Due to this, they exhibit different patterns in their structure and dynamics, commonly known as *collective behavior*. While various properties can characterize a social system's building units, only a few enforce collective behavior in the systems. The phenomenon is known as *universality* in physics and is commonly observed in social systems such as in voting behavior [2], or scientific citations [3]. It indicates the existence of the universal mechanisms that govern the dynamics of the system [1].

Social groups, informal or formal, are mesoscopic building elements of every socio-economic system that direct its emergence, evolution, and disappearance [4]. The examples span from countries, economies, and science to society. Settlements, villages, towns, and cities are formal and highly structured social groups of countries. Their organization and growth determine the functioning and sustainability of every society [5]. Companies are the building blocks of an economic system, and their dynamics are essential indicators of the level of its development [6]. Scientific conferences, as scientific groups, enable fast dissemination of the latest results, exchange, and evaluation of ideas as well as a knowledge extension, and thus are an integral part of science [7]. The membership of

https://doi.org/10.1088/1742-5468/aca0e9

individuals in various social groups, online and offline, can be essential when it comes to the quality of their life [8–10]. Therefore, it is not surprising that the social group emergence and evolution are at the center of the attention of many researchers [11–14].

The availability of large-scale and long-term data on various online social groups has enabled the detailed empirical study of their dynamics. The focus was mainly on the individual groups and how structural features of social interaction influence whether individuals will join the group [15] and remain its active members [7, 16]. The study on LiveJournal [15] groups has shown that decision of an individual to join a social group is greatly influenced by the number of her friends in the group and the structure of their interactions. The conference attendance of scientists is mainly influenced by their connections with other scientists and their sense of belonging [7]. The sense of belonging of an individual in social groups is achieved through two main mechanisms [16]: expanding the social circle at the beginning of joining the group and strengthening the existing connections in the later phase. Analysis of the evolution of large-scale social networks has shown that edge locality plays a critical role in the growth of social networks [17]. The dynamics of social groups depend on their size [18]. Small groups are more cohesive with continued long-term, while large groups change their active members constantly [18]. These findings help us understand the growth of a single group, the evolution of its social network, and the influence of the network structure on group growth. However, how the growth mechanisms influence the distribution of members of one social system among groups is yet to be understood.

Furthermore, it is not clear whether the growth mechanisms of social groups are universal or system-specific. The size distribution of social groups has not been extensively studied. Rare empirical evidence of the size distribution of social groups indicates that it follows power-law behavior [19]. However, the distribution of company sizes follows log-normal behavior and remains stable over decades [20, 21]. Analysis of the cities' sizes shows that all cities' distribution also follows a log-normal distribution [22]. In contrast, the distribution of the largest cities resembles Zipf's distribution [23].

A related question that should be addressed is whether we can create a unique yet relatively simple microscopic model that reproduces the distribution of members between groups and explains the differences observed between social systems. French economist Gibrat proposed a simple growth model to produce companies' and cities' observed log-normal size distribution. However, the analysis of the growth rate of the companies [20] has shown that growth mechanisms are different from those assumed by Gibrat. In addition, the analysis of the growth of the online social networks showed that the population size and spatial factors do not determine population growth, and it deviates from Gibrat's law [24]. Other mechanisms, for instance, growth through diffusion, have been used to model and predict rapid group growth [25]. However, the growth mechanisms of various social groups and the source of the scaling observed in socio-economic systems remain hidden.

Here we analyze the size distribution of formal social groups in three data sets: Meetup groups based in London and New York and subreddits on Reddit. We are interested in the scaling behavior of size distributions and the distribution of growth rates. Empirical analysis of the dependence of growth rates, shown in this work, indicates that growth cannot be explained through Gibrat's model. Here we contribute with a simple microscopic model that incorporates some of the findings of previous research [15, 19]. We show that the model can reproduce size and growth rate distributions for both studied systems. Moreover, the model is flexible and can produce a broad set of log-normal size distributions depending on the value of model parameters.

The paper is organized as follows: in section 2 we describe the data, while in section 3 we present our empirical results. In section 4 we introduce model parameter and principles. In section 5 we demonstrate that model can reproduce the growth of social groups in both systems and show the results for different values of model parameters. Finally, in section 6, we present concluding remarks and discuss our results.

2. Data

We analyze the growth of social groups from two widely used online platforms: Reddit and Meetup. Reddit[3] enables sharing of diverse web content, and members of this platform interact exclusively online through posts and comments. The Meetup[4] allows people to use online tools to organize offline meetings. The building elements of the Meetup system are topic-focused groups, such as food lovers or data science professionals. Due to their specific activity patterns—events where members meet face-to-face—Meetup groups are geographically localized, and interactions between members are primarily offline.

We compiled the Reddit data from https://pushshift.io/. This site collects data daily and, for each month, publishes merged comments and submissions in the form of JSON files. Specifically, we focus on subreddits—social groups of Reddit members interested in a specific topic. We selected subreddits created between 2006 and 2011 that were active in 2017 and followed their growth from their beginning until 2011. The considered dataset contains 17073 subreddits with 2195 677 active members, with the oldest originating from 2006 and the youngest being from 2011. For each post under a subreddit, we extracted the information about the member-id of the post owner, subreddit-id, and timestamp. As we are interested in the subreddits growth in the number of members, for each subreddit and member-id, we selected the timestamp when a member made a post for the first time. Finally, in the dataset, we include only subreddits active for at least two months.

The Meetup data were downloaded in 2018 using public API. The Meetup platform was launched in 2003, and when we accessed the data, there were more than 240 000 active groups. For each group, we extracted information about the date it had been founded, its location, and the total number of members. We focused on the groups founded in a period between 2003 and 2017 in big cities, London and New York, where the Meetup platform achieved considerable popularity. We considered groups active for at least two months. There were 4673 groups with 831 685 members in London and 4752 groups with 1059 632 members in New York. In addition, we extracted the ids of group

[3] https://reddit.com/.

[4] www.meetup.com.

members, the information about organized events, and which members attended these events. Based on this, we obtained the date when a member joined a group, the first time she participated in a group event.

For all systems, we extracted the timestamp when the member joined the group. Each data set has a form (u_{id}, g_{id}, t_i), representing the connection between users and groups. When the system has two separate partitions, the natural extension is a bipartite network where links are drawn between nodes of different sets, indicating the user's memberships. The degree of group nodes is exactly the group size. Having the temporal component in data, we can follow the evolution of the network. Based on this information, we can calculate the number of new members per month $N_i(t)$, the group size $S_i(t)$ at each time step, and the growth rate for each group. The time step for all three data sets is one month. The size of the group i at time step t is the number of members that joined that group ending with the month, i.e. $S_i(t) = \sum_{k=t_{i0}}^{k=t} N_i(t)$, where t_{i0} is the time step in which the group i was created. Once the member joins the group, it has an active status by default, which remains permanent. For these reasons, the size of considered groups is a non-decreasing function. The growth rate $R_i(t)$ at step i is obtained as logarithm of successive sizes $R_i(t) = \log(S_i(t)/S_i(t-1))$.

While the forms of communication between members and activities that members engage in differ for considered systems, some common properties exist between them. Members can form new groups and join the existing ones. Furthermore, each member can belong to an unlimited number of groups. For these reasons, we can use the same methods to study and compare the formation of groups on Reddit and Meetup.

3. Empirical analysis of social group growth

Figure 1 summarizes the properties of the groups in Meetup and Reddit systems. The number of groups grows exponentially over time. Nevertheless, we notice that Reddit has a substantially larger number of groups than Meetup. The Reddit groups are prone to engage more members in a shorter period. The size of the Meetup groups ranges from several members up to several tens of thousands of members, while sizes of subreddits are between a few tens of members up to several million. The distributions of normalized group sizes follow the log-normal distribution (see table S1 and figure S1 in SI)

$$P(S) = \frac{1}{\frac{S}{S_0}\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\ln\left(\frac{S}{S_0}\right) - \mu\right)^2}{2\sigma^2}\right), \tag{1}$$

where S is the group size, S_0 is the average group size in the system, and μ and σ are parameters of the distribution. We used *power-law* package [26] to fit equation (1) to empirical data and found that distribution of groups sizes for Meetup groups in London and New York follow similar distributions with the values of parameters $\mu = -0.93$, $\sigma = 1.38$ and $\mu = -0.99$ and $\sigma = 1.49$ for London and New York respectively. The distribution of sizes of subreddits also has the log-normal shape with parameters $\mu = -5.41$ and $\sigma = 3.07$.

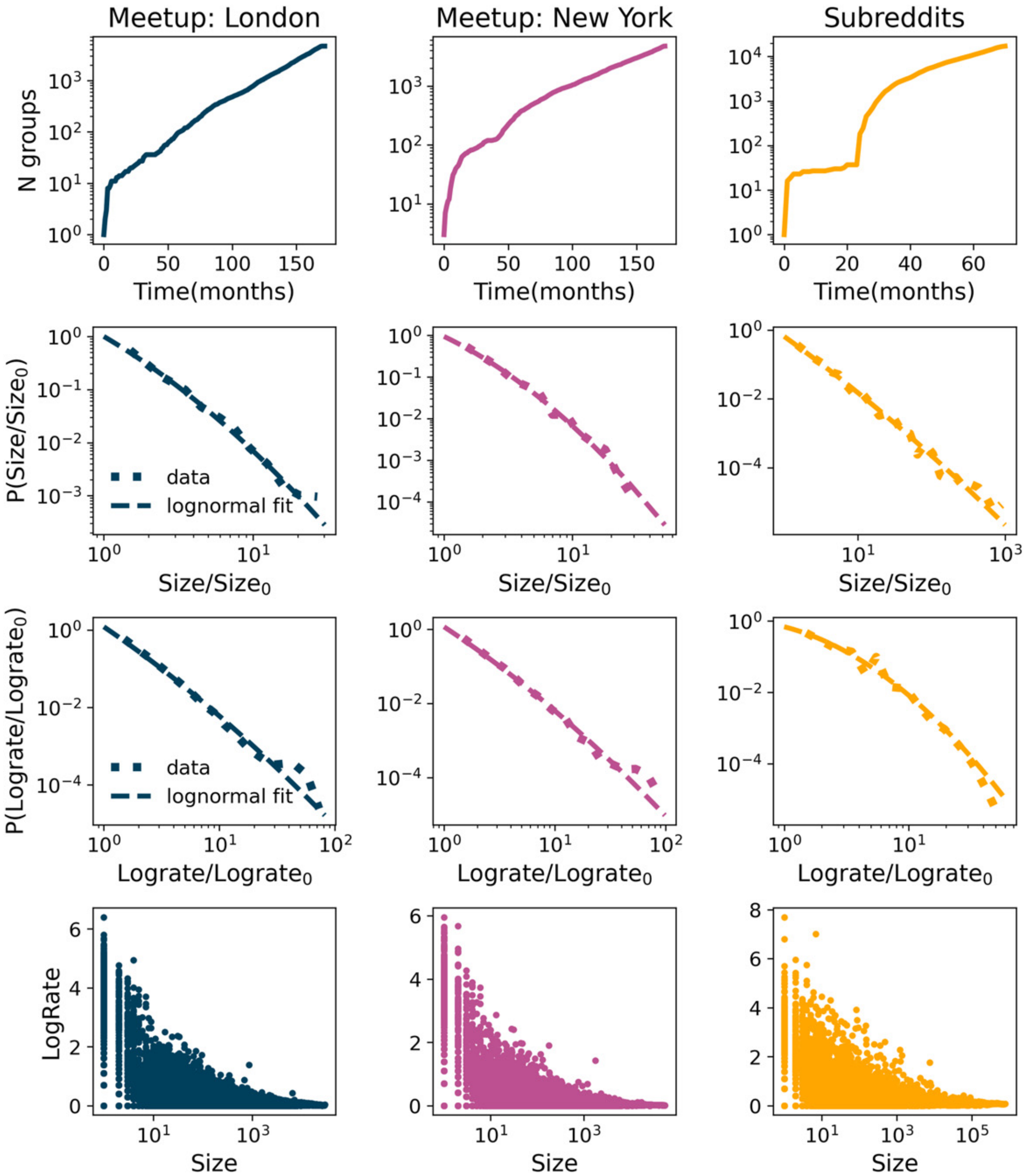

Figure 1. The number of groups over time, normalized sizes distribution, normalized log-rates distribution and dependence of log-rates and group sizes for Meetup groups created in London and New York and subreddits. The number of groups grows exponentially over time, while the group size distributions, and log-rates distributions follow log-normal. Lograte depend on the size of the group, implying that the growth cannot be explained by Gibrat law.

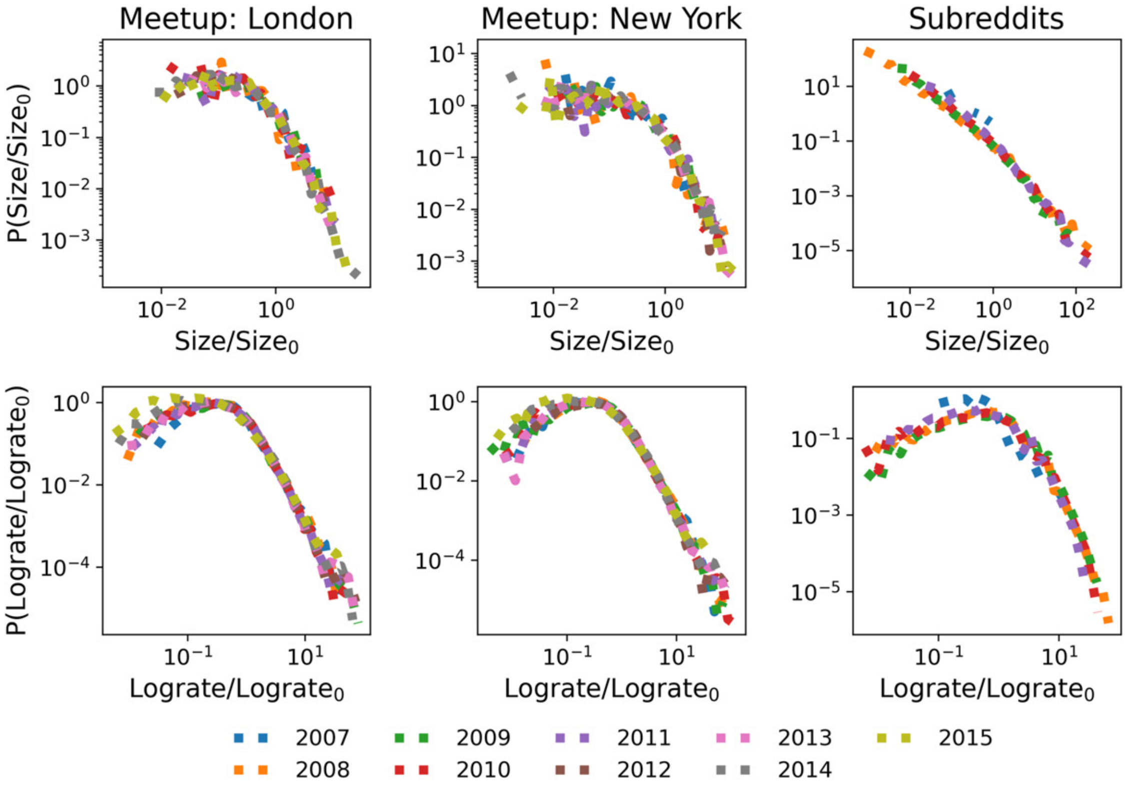

Figure 2. The figure shows the groups' sizes distributions and log-rates distributions. Figures in the top panels show the distribution of normalized sizes of groups created in the same year. Distributions for the same system and different years follow same log-normal distribution indicating existence of universal growth patterns.

Multiplicative processes can generate the log-normal distributions [27]. If there is a quantity with size $S_i(t)$ at time step t, it will grow so after time period δ the size of the quantity is $S(t + \Delta t) = S(t)r$, where r represents a random number. The Gibrat law states that growth rates r are uncorrelated and do not depend on the current size. To describe the growth of social groups, we calculate the logarithmic growth rates $R_i(t)$. According to Gibrat law the distribution of logarithmic growth rates is normal, or, as it is shown in many studies, it is better explained with Laplacian ('tent-shaped') distribution [28, 29]. In figure 1 we show the distributions of log-rates for all three data sets. Log-rates are very well approximated with a log-normal distribution. Furthermore, the bottom panels of figure 1 show that log-rates are not independent of group size. Figure 1 shows that these findings imply that the growth of Meetup and Reddit groups violates the basic assumptions of Gibrat's law [30, 31] and that it cannot be explained as a simple multiplicative process.

We are considering a relatively significant period for online groups. The fast expansion of information communications technologies (ICT) changed how members access online systems. With the use of smartphones, online systems became more available, which led to the exponential growth of ICTs systems and potential change in the mechanisms that influence the social groups' growth. For these reasons, we aggregate groups according to the year they were founded for each of the three data sets and look at the distributions of their sizes at the end of 2017 for Meetup groups and 2011 for Reddit. For each year and each of the three data sets, we calculate the average size of the groups created in a year $y\langle S^y\rangle$. We normalize the size of the groups originating in year y with the corresponding average size $s_i^y = S_i^y/\langle S^y\rangle$ and calculate the distribution of the normalized sizes for each year. The distribution of normalized sizes for all years and data sets is shown in figure 2. All distributions exhibit log-normal behavior. Furthermore, the distributions for the same data set and different years follow a universal curve with the same value of parameters μ and σ. The universal behavior is observed for the distribution of normalized log-rates as well, see figure 2 (bottom panels). These results indicate that the growth of the social groups did not change due to the increased growth of members in systems. Furthermore, it implies that the growth is independent of the size of the whole data set.

4. Model

The growth of social groups cannot be explained by the simple rules of Gibrat's law. Previous research on group growth and longevity has shown that social connections with members of a group influence individual's choice to join that group [19, 25]. Individuals' interests and the need to discover new content or activity also influence the diffusion of individuals between groups. Furthermore, social systems constantly grow since new members join every minute. The properties of the growth signal that describes the arrival of new members influence both dynamics of the system [32, 33] and the structure of social interactions [34]. The number of social groups in the social systems is not constant. They are constantly created and destroyed.

In [19], the authors propose the co-evolution model of the growth of social networks. In this model, the authors assume that the social system evolves through the co-evolution of two networks: a network of social contacts between members and a network of members' affiliations with groups. This model addresses the problem of the growth of social networks that includes both linking between members and social group formation. In this model, a member of a social system selects to join a group either through random selection or according to her social contacts. In the case of random selection, there is a selection preference for larger groups. If a member chooses to select a group according to her social contacts, the group is selected randomly from the list of groups with which her friends are already affiliated.

In [19], the authors demonstrate that mechanisms postulated in the model could reproduce the power-law distribution of group sizes observed for some social networks. However, as illustrated in section 3, the distribution of group sizes in real systems is not necessarily power-law. Our rigorous empirical analysis shows that the distribution of social group sizes exhibits log-normal behavior. To fill the gap in understanding how social groups in the social system grow, we propose a model of group growth that

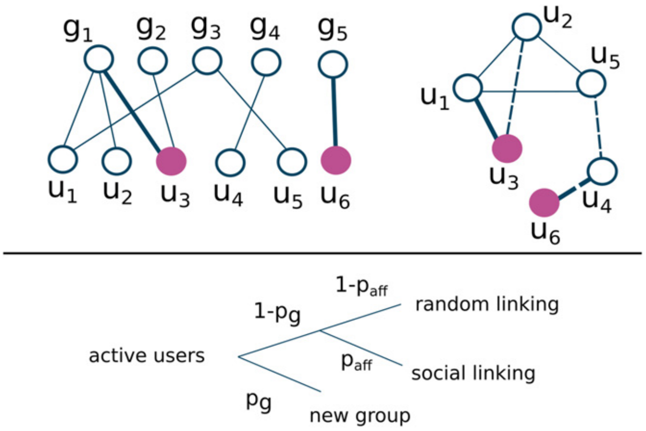

Figure 3. The top panel shows bipartite (member-group) and social (member-member) network. Filled nodes are active members, while thick lines are new links in this time step. In the social network dashed lines show that members are friends but still do not share same groups. The lower panel shows model schema. Example: member u_6 is a new member. First it will make random link with node u_4, and then with probability p_g makes new group g_5. With probability p_a member u_3 is active, while others stay inactive for this time step. Member u_3 will with probability $1 - p_g$ choose to join one of old groups and with probability p_{aff} linking is chosen to be social. As its friend u_2 is member of group g_1, member u_3 will also join group g_1. Joining group g_1, member u_3 will make more social connections, in this case it is member u_1.

combines random and social diffusion between groups but follows different rules than the co-evolution model [19].

Figure 3 shows a schematic representation of our model. Similar to the co-evolution model [19], we represent a social system with two evolving networks, see figure 3. One network is a bipartite network that describes the affiliation of individuals to social groups $\mathcal{B}(V_U, V_G, E_{UG})$. This network consists of two partitions, members V_U and groups V_G, and a set of links E_{UG}, where a link $e(u, g)$ between a member u and a group g represents the member's affiliation with that group. Bipartite network grows through three activities: the arrival of new members, the creation of new groups, and members joining groups. In bipartite networks, links only exist between nodes belonging to different partitions. However, as we explained above, social connections affect whether a member will join a certain group or not. In the simplest case, we could assume that all members belonging to a group are connected. However, previous research on this subject [15, 16, 19] has shown that the existing social connections of members in a social group are only a subset of all possible connections. For these reasons, we introduce another network $\mathcal{G}(V_U, E_{UU})$ that describes social connections between members. The social network grows by adding new members to the set V_U and creating new links between them. The member partition in bipartite network $\mathcal{B}(V_U, V_G, E_{UG})$ and set of nodes in members' network $\mathcal{G}(V_U, E_{UU})$ are identical.

For convenience, we represent the bipartite and social network of members with adjacency matrices B and A. The element of the matrix B_{ug} equals one if member u

is affiliated with group g, and zero otherwise. In matrix A, the element $A_{u_1u_2}$ equals one if members u_1 and u_2 are connected and zero otherwise. The neighborhood $\mathcal{N}_u$ of member u is a set of groups with which the member is affiliated. On the other hand, the neighborhood $\mathcal{N}_g$ of a group g is a set of members affiliated with that group. The size S_g of set $\mathcal{N}_g$ equals to the size of the group g.

In our model, the time is discrete, and networks evolve through several simple rules. In each time step, we add $N_U(t)$ new members and increase the size of the set V_U. For each newly added member, we create the link to a randomly chosen old member in the social network G. This condition allows each member to perform social diffusion [25], i.e. to select a group according to her social contacts. Not all members from setting V_U are active in each time step. Only a subset of existing members is active in each time step. The activity of old members is a stochastic process determined by parameter p_a; every old member is activated with probability p_a. Old members are activated in this way, and new members make a set of active members $\mathcal{A}_U$ at time t.

The group partition V_G grows through creating new groups. Each active member $u \in \mathcal{A}_U$ can decide with probability p_g to create a new group or to join an already existing one with probability $1 - p_g$.

If the active member u decides that she will join an existing group, she first needs to choose a group. A member u with probability p_{aff} decides to select a group based on her social connections. For each active member, we look at how many social contacts she has in each group. The number of social contacts s_{ug} that member u has in the group g equals the overlap of members affiliated with a group g and social contacts of member u, and is calculated according to

$$s_{ug} = \sum_{u_1 \in \mathcal{N}_g} A_{uu_1}. \tag{2}$$

Member u selects an old group g to join according to probability P_{ug} that is proportional to s_{ug}. Member-only considers groups with which it has no affiliation. However, if an active member decides to neglect her social contacts in the choice of the social group, she will select a random group from the set V_G with which she is not yet affiliated.

After selecting the group g, a member joins that group, and we create a link in the bipartite network between a member u and a group g. At the same time, the member selects X members of a group g which do not belong to her social circle and creates social connections with them. As a consequence of this action, we make X new links in-network $\mathcal{G}$ between member u and X members from a group g.

The evolution of bipartite and social networks, and consequently growth of social groups, is determined by parameters p_a, p_g and p_{aff}. Parameter p_a determines the activity level of members and takes values between 0 and 1. Higher values of p_a result in a higher number of active members and thus faster growth of the number of links in both networks and the size and number of groups. Parameter p_g in combination with parameter p_a determines the growth of the set V_G. $p_g = 1$ means that members only create new groups, and the existing network consists of star-like subgraphs with members being central nodes and groups as leaves. On the other hand, $p_g = 0$ means that there is no creation of new groups, and the bipartite network only grows through adding new members and creating new links between members and groups.

Parameter p_{aff} determines the importance of social diffusion. $p_{\mathrm{aff}} = 0$ means that social connections are irrelevant, and the group choice is random. On the other hand, $p_{\mathrm{aff}} = 1$ means that only social contacts become important for group selection.

Several differences exist between the model presented in this work and the co-evolution model [19]. In our model, p_{aff} is constant and the same for all members. In the co-evolution model, this probability depends on members' degrees. The members are activated in our model with probability p_a. In contrast, in the co-evolution model, members are constantly active from the moment they are added to a set V_U until they become inactive after time t_a. Time t_a differs for every member and is drawn from an exponential distribution. In the co-evolution model, the number of social contacts members have within the group is irrelevant to its selection. On the other hand, in our model, members tend to choose groups more often in which there is a greater number of social contacts. While in our model, in the case of a random selection of a group, a member selects with equal probability a group that she is not affiliated with, in the co-evolution model, the choice of group is preferential.

5. Results

The distribution of group sizes produced by our and co-evolution models significantly differ. The distribution of group sizes in the co-evolution model is a power-law. Our model enables us to create groups with log-normal size distribution and expand classes of social systems that can be modeled.

5.1. Model properties

First, we explore the properties of size distribution depending on parameters p_g and p_{aff}, for the fixed value of activity parameter p_a and constant number of members added in each step $N(t) = 30$. When the group is created, its size $S(t_0) = 1$, so the group creator cannot make new social connections until new members arrive. While a group has less than X members, new users will make social connections with all available members in the group. After the group size reaches the threshold of X members, a new user creates X connections. Our detailed analysis of the results for different parameter values X shows that these results are independent of their value. We set the value of parameter X to 25 for all simulations presented in this work. Our detailed analysis of the results for different parameter values X shows that these results are independent of their value.

Figure 4 shows some of the selected results and their comparison with power-law and log-normal fits. We see that values of both p_g and p_{aff} parameters, influence the type and properties of size distribution. For low values of parameter p_g, left column in figure 4, the obtained distribution is log-normal. The width of the distribution depends on p_{aff}. Higher values of p_{aff} lead to a broader distribution.

As we increase p_g, right column in figure 4, the size distribution begins to deviate from log-normal distribution. The higher the value of parameter p_g, the total number of groups grows faster. For $p_g = 0.5$, half of the active members in each time step create a group, and the number of groups increases fast. How members are distributed in these groups

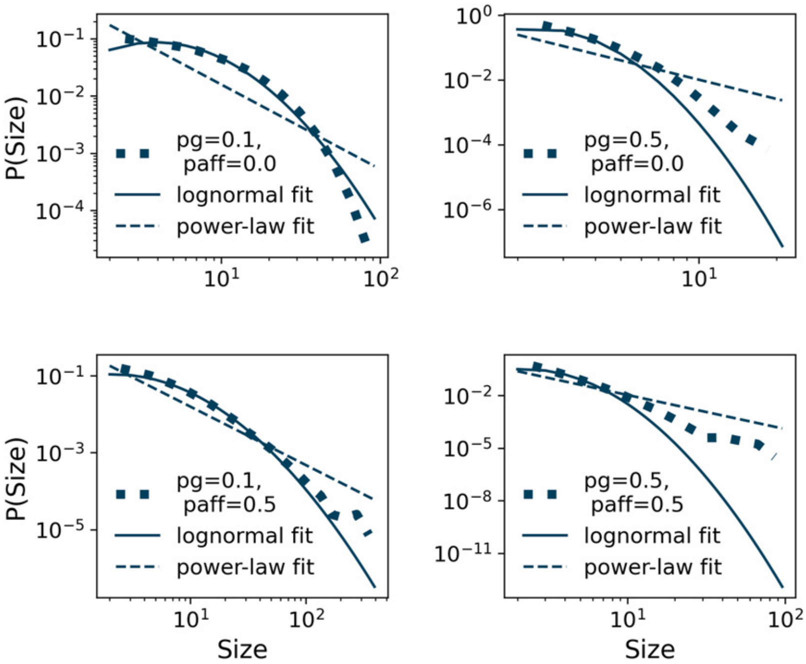

Figure 4. The distribution of sizes for different values of p_g and p_{aff} and constant p_a and growth of the system. The combination of the values of parameters of p_g and p_{aff} determine the shape and the width of the distribution of group sizes.

depends on the parameter p_{aff} value. When $p_{\mathrm{aff}} = 0$, social connections are irrelevant to the group's choice, and members select groups randomly. The obtained distribution slightly deviates from log-normal, especially for large group sizes. In this case, large group sizes become more probable than in the case of the log-normal distribution. The non-zero value of parameter p_{aff} means that the choice of a group becomes dependent on social connections. When a member chooses a group according to her social connections, larger groups have a higher probability of being affiliated with the social connections of active members, and thus this choice resembles preferential attachment. For these reasons, the obtained size distribution has more broad tail than log-normal distribution and begins to resemble power-law distribution.

The top panel of figure S3 in SI shows how the shape of distribution is changing with the value of parameter p_{aff} and fixed values of $p_a = 0.1$ and $p_g = 0.1$. Preferential selection groups according to their size instead of one where a member selects a group with equal probability leads to a drastic change in the shape of the distribution, bottom panel figure S3 in SI. As is to be expected, the distribution of group sizes with preferential attachment follows power-law behavior.

5.2. Modeling real systems

The social systems do not grow at a constant rate. In [34], the authors have shown that features of growth signal influence the structure of social networks. For these reasons, we use the real growth signal from Meetup groups located in London and New York

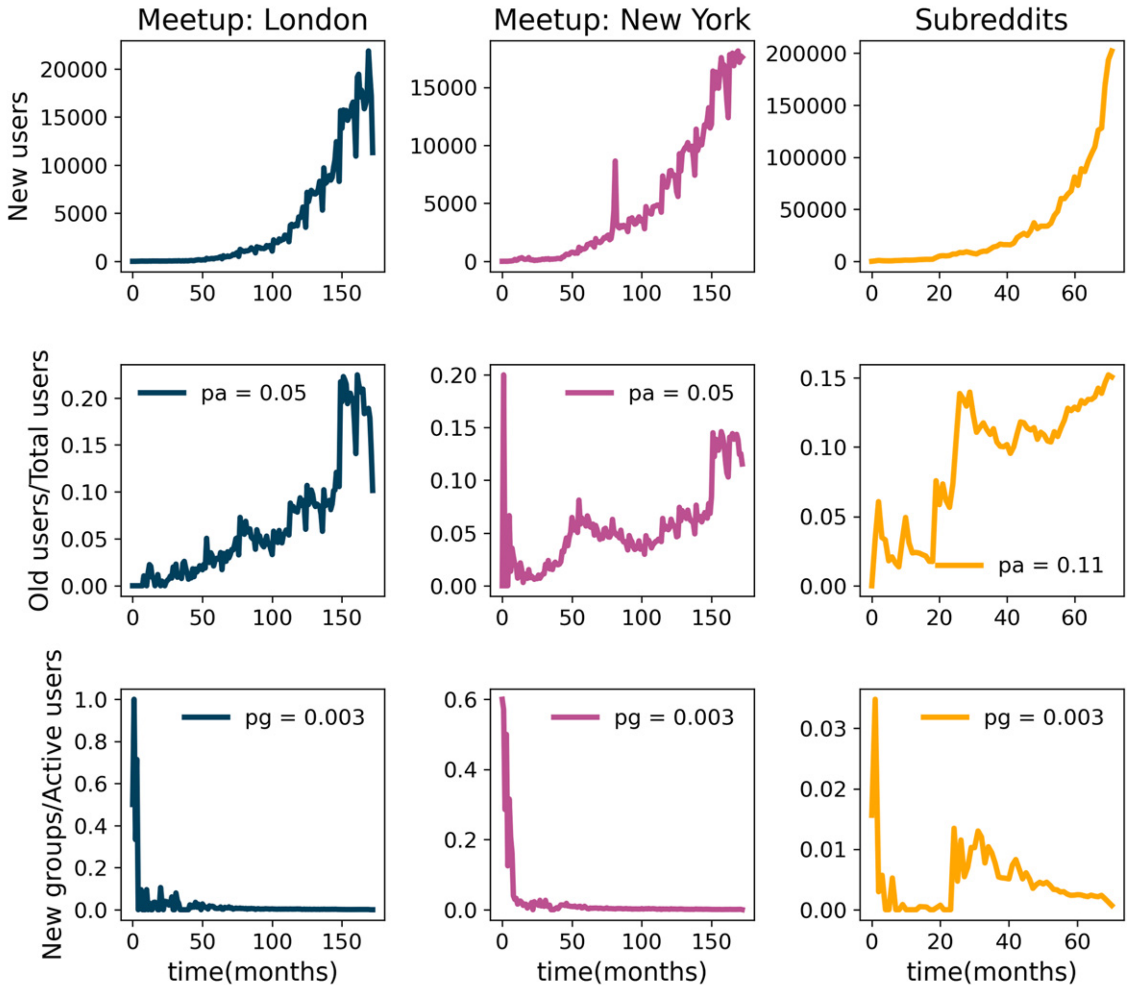

Figure 5. The time series of the number of new members (top panels). The time series of the ratio between several old active members and total members in the system (middle panels); its median value approximates the parameter p_a, the probability that the user is active. The bottom panels show the time series of the ratio between new groups and active members; its median value approximates the probability that active users create a new group, parameter p_g.

and Reddit to simulate the growth of the social groups in these systems. Figure 5 (top) shows the time series of the number of new members that join each of the considered systems each month. All three data sets have relatively low growth at the beginning, and then the growth accelerates as the system becomes more popular.

We also use empirical data to estimate p_a, p_g and p_{aff}. The data can approximate the probability that old members are active p_a and that new groups are created p_g. Activity parameter p_a is the ratio between the number of old members active in month t and the total number of members in the system at time t. Figure 5 (middle) shows the variation of parameter p_a during the considered time interval for each system. The value of this parameter fluctuates between 0 and 0.2 for London and New York based Meetup

Table 1. Jensen Shannon divergence between group sizes distributions from model and data. In the model we vary affiliation parameter p_{aff} and find its optimal value (bold text).

p_{aff}	JS cityLondon	JS cityNY	JS reddit2012
0.1	0.0161	0.0097	0.002 41
0.2	0.0101	0.0053	0.002 05
0.3	0.0055	0.0026	0.001 59
0.4	0.0027	**0.0013**	0.001 04
0.5	**0.0016**	0.0015	0.000 74
0.6	0.0031	0.0035	0.000 48
0.7	0.0085	0.0081	0.000 39
0.8	0.0214	0.0167	**0.000 34**
0.9	0.0499	0.0331	0.000 47

groups, while its value is between 0 and 0.15 for Reddit. To simplify our simulations, we assume that p_a is constant in time and estimate its value as its median value during the 170 months for Meetup and 80 months for Reddit systems. For Meetup groups based in London and New York $p_a = 0.05$, while Reddit members are more active on average and $p_a = 0.11$ for this system.

Figure 5 bottom row shows the evolution of parameter p_g for the considered systems. The p_g in month t is estimated as the ratio between the groups created in month $tNg_{\mathrm{new}}(t)$ and the total number of groups in that month $Ng_{\mathrm{new}}(t) + Ng_{\mathrm{old}}(t)$, i.e. $p_g(t) = \frac{Ng_{\mathrm{new}}(t)}{N_{\mathrm{new}}(t)+N_{\mathrm{old}}(t)}$. We see from figure 5 that $p_g(t)$ has relatively high values at the beginning of the system's existence. This is not surprising. Initially, these systems have a relatively small number of groups and often cannot meet the needs of the content of all their members. As the time passes, the number of groups and content scope within the system grows, and members no longer have a high need to create new groups. Figure 5 shows that p_g fluctuates less after the first few months, and thus we again assume that p_g is constant in time and set its value to the median value during 170 months for Meetup and 80 months for Reddit. For all three systems p_g has the value of 0.003.

The affiliation parameter p_{aff} cannot estimate directly from the empirical data. For these reasons, we simulate the growth of social groups for each data set with the time series of new members obtained from the real data and estimated values of parameters p_a and p_g, while we vary the value of p_{aff}. We compare the distribution of group sizes obtained from simulations for different values of p_{aff} with ones obtained from empirical analysis using Jensen Shannon (JS) divergence. The JS divergence [35] between two distributions P and Q is defined as

$$JS(P,Q) = H\left(\frac{P+Q}{2}\right) - \frac{1}{2}(H(P) + H(Q)) \qquad (3)$$

where $H(p)$ is Shannon entropy $H(p) = \sum_x p(x)\log(p(x)$. The JS divergence is symmetric and if P is identical to Q, $JS = 0$. The smaller the value of JS divergence, the better is the match between empirical and simulated group size distributions. Table 1

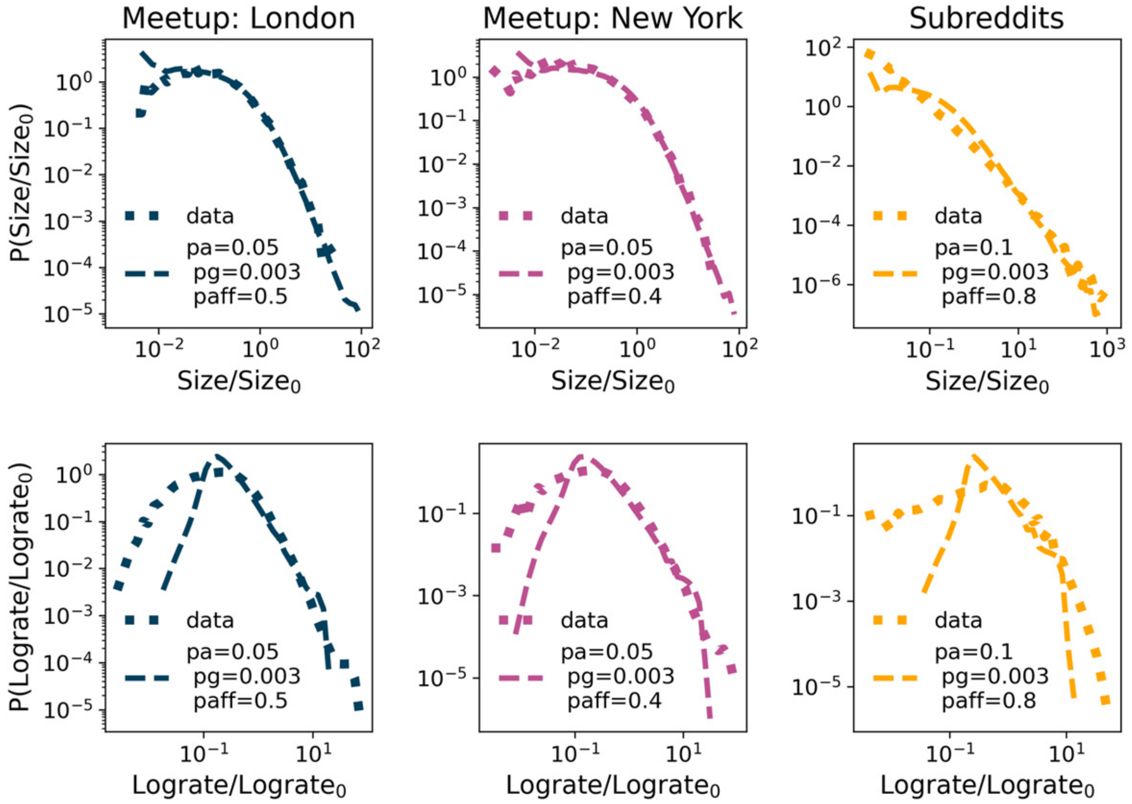

Figure 6. The comparison between empirical and simulation distribution for group sizes (top panels) and log-rates (bottom panels).

shows the value of JS divergence for all three data sets. We see that for London based Meetup groups the affiliation parameter is $p_{\text{aff}} = 0.5$, for New York groups $p_{\text{aff}} = 0.4$, while the affiliation parameter for Reddit $p_{\text{aff}} = 0.8$. Our results show that social diffusion is important in all three data sets. However, Meetup members are more likely to join groups at random, while for the Reddit members their social connections are more important when it comes to choice of the subreddit.

Figure 6 compares the empirical and simulation distribution of group sizes for considered systems. We see that empirical distributions for Meetup groups based in London and New York are well reproduced by the model and chosen values of parameters. In the case of Reddit, the distribution is broad, and the model reproduces the tail of the distribution well. Figure S2 and table S2 in SI confirm that the distribution of group sizes follow a log-normal distribution.

The bottom row of figure 6 shows the distribution of logarithmic values of growth rates of groups obtained from empirical and simulated data. We see that the tails of empirical distributions for all three data sets are well emulated by the ones obtained from the model. The deviations we observe are the most likely consequence of using median values of parameters p_a, p_g, and p_{aff}.

6. Discussion and conclusions

The results of empirical analysis show that there are universal growth rules that govern the growth of social systems. We analysed the growth of social groups for three data sets, Meetup groups located in London and New York and Reddit. We showed that the distribution of group sizes has log-normal behaviour. The empirical distributions of normalised sizes of groups created in different years in a single system fall on top of each other, following the same log-normal distributions. Due to a limited data availability, we only study three data sets which may affect the generality of our results. However, the substantial differences between Reddit and Meetup social systems when it comes to their popularity, size and purpose, demonstrate that observed growth patterns are universal.

Even though the log-normal distribution of group sizes can originate from the proportional growth model, Gibrat law, we show that it does not apply to the growth of online social groups. The monthly growth rates are log-normally distributed and dependent on the size of a group. Gibrat law was proposed to describe the growth of various socio-economical systems, including the cities and firms. Recent studies showed that the growth of cities and firms [21, 36, 37] goes beyond Gibrat law. Still, our findings confirm the existence of universal growth patterns, indicating the presence of the general law in the social system's growth.

While the growth of the social groups does not follow the Gibrat law, one could ask whether there are other simple models of social group growth. The basic growth model underlying any log-normal distribution is a multiplicative process. The size of the system in time t is equal to its size in time $t-1$ multiplied by some factor. In our case, where the groups only grow and do not shrink, the factor has to be larger than one. When we model the growth of real social groups, we need to take into account several factors: (1) social systems grow through the addition of new members; (2) the number of social groups is not constant, it grows with time; (3) one person can be a member of multiple groups at the same time. The simplest model that considers all three factors but disregards social factors, and thus a network structure, would be the one where members randomly choose the groups they will join. The described situation is an extreme case of our model with $p_{\text{aff}} = 0$, see figure 4, top left panel. By setting the values of $p_{\text{aff}} = 0$ and taking the value of $N(t)$ and p_g as an estimate from real data, we can reproduce a log-normal distribution with parameters that do not match empirical data, see table 1. While the distributions of group size in different systems follow log-normal behavior, the parameters of these distributions differ from system to system. This indicates the existence of additional factors in the multiplicative process that govern multiplicative growth. The network effect is crucial in explaining many instances of collective social dynamics, including the person's choice to join a certain group [14]. Here we show that members' diffusion between groups governed by social influence allows us to use the same model to explain the growth of groups in different social systems by tuning its importance.

The model proposed in [19] is able to produce only power-law distributions of group sizes. However, our empirical analysis shows that these distributions can also have a log-normal behavior. Thus, we propose a new model that emulate log-normal distributions. The analysed groups grow through two mechanisms [19]: members join a group that is chosen according to their interests or by social relations with the group's members. The number of members in the system is growing as well as the number of groups. While the processes that govern the growth of social groups are the same, their importance varies among the systems. The distributions for Meetup groups located in the London and New York have similar log-normal distribution parameter values, while for Reddit, the distribution is broader. Numerical simulations further confirm these findings. Different modalities of interactions between their members can explain the observed differences.

Meetup members need to invest more time and resources to interact with their peers. The events are localised in time and space, and thus the influence of peers in selecting another social group may be limited. On the other hand, Reddit members do not have these limitations. The interactions are online, asynchronous, and thus not limited in time. The influence of peers in choosing new subreddits and topics thus becomes more important. The values of p_{aff} parameters for Meetup and Reddit imply that social connections in diffusion between groups are more critical in Reddit than in Meetup.

The purpose of the research presented in this paper was to provide a model of social group growth that can reproduce the log-normal distribution of group sizes in different systems. The model is based on bipartite network dynamics allowing us to study other network properties and compare them to empirical data. The empirical data are limited and only contain explicit information about the connections between groups and their members. The distribution of group sizes is the exact degree distribution of the group partition. We show that these properties are reproduced with our model, see figure 6. When it comes to the degree distribution of members, that is, the number of groups a member is affiliated with, our model does not reproduce this distribution. The number of groups a member is affiliated to is equal to number of her activities. The activity of a member is controlled with probability p_a. In our model, the probability p_a is equal for all members, and thus the emerging degree distribution is exponential [38]. We do not study the properties of the members' partitions in detail, as our focus is on the growth of groups' partitions and mechanisms that influence the members' choice to join the groups. On the other hand, studying how groups are distributed among members could give us insight into what motivates members to be active. Previous work proposed that each member has a lifetime [17], but different linking rules could be considered; for example, p_a could be preferential toward high-degree members, and the age or even social connections of members could be relevant.

The results presented in this paper contribute to our knowledge of the growth of socio-economical systems. The previous study analysed the social systems in which size distributions follow the power-law, which is the consequence of a preferential choice of groups during the random diffusion of members. Our findings show that preferential selection of groups during social diffusion and uniform selection during random diffusion result in log-normal distribution of groups sizes. Furthermore, we show that broadness of the distribution depends on the involvement of social diffusion in the growth process. Our model increases the number of systems that can be modelled and help us better understand the growth and segmentation of social systems and predict their evolution.

Acknowledgments

We acknowledge funding provided by the Institute of Physics Belgrade, through the grant by the Ministry of Education, Science, and Technological Development of the Republic of Serbia. Numerical simulations were run on the PARADOX-IV supercomputing facility at the Scientific Computing Laboratory, National Center of Excellence for the Study of Complex Systems, Institute of Physics Belgrade.

References

[1] Castellano C, Fortunato S and Loreto V 2009 Statistical physics of social dynamics *Rev. Mod. Phys.* **81** 591
[2] Chatterjee A, Mitrović M and Fortunato S 2013 Universality in voting behavior: an empirical analysis *Sci. Rep.* **3** 1–9
[3] Radicchi F, Fortunato S and Castellano C 2008 Universality of citation distributions: toward an objective measure of scientific impact *Proc. Natl Acad. Sci. USA* **105** 17268–72
[4] Firth R 2013 *Elements of Social Organisation* (London:Routledge)
[5] Barthelemy M 2016 *The Structure and Dynamics of Cities* (Cambridge: Cambridge University Press)
[6] Hidalgo C A and Hausmann R 2009 The building blocks of economic complexity *Proc. Natl Acad. Sci. USA* **106** 10570–5
[7] Smiljanić J, Chatterjee A, Kauppinen T and Dankulov M M 2016 A theoretical model for the associative nature of conference participation *PLoS One* **11** e0148528
[8] Montazeri A, Jarvandi S, Haghighat S, Vahdani M, Sajadian A, Ebrahimi M and Haji-Mahmoodi M 2001 Anxiety and depression in breast cancer patients before and after participation in a cancer support group *Patient Educ. Counseling* **45** 195–8
[9] Davison K P, Pennebaker J W and Dickerson S S 2000 Who talks? The social psychology of illness support groups *Am. Psychol.* **55** 205
[10] Cho W K T *et al* 2012 The tea party movement and the geography of collective action *Q. J. Pol. Sci.* **7** 105–33
[11] Aral S and Walker D 2012 Identifying influential and susceptible members of social networks *Science* **337** 337–41
[12] González-Bailón S, Borge-Holthoefer J and Moreno Y 2013 Broadcasters and hidden influentials in online protest diffusion *Am. Behav. Sci.* **57** 943–65
[13] Török J, Iñiguez G, Yasseri T, San Miguel M, Kaski K and Kertész J 2013 Opinions, conflicts, and consensus: modeling social dynamics in a collaborative environment *Phys. Rev. Lett.* **110** 088701
[14] Yasseri T, Sumi R, Rung A, Kornai A and Kertész J 2012 Dynamics of conflicts in wikipedia *PLoS One* **7** e38869
[15] Backstrom L, Huttenlocher D, Kleinberg J and Lan X 2006 Group formation in large social networks: membership, growth, and evolutionProc. 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining pp 44–54
[16] Smiljanić J and Dankulov M M 2017 Associative nature of event participation dynamics: a network theory approach *PLoS One* **12** e0171565
[17] Leskovec J, Backstrom L, Kumar R and Tomkins A 2008 Microscopic evolution of social networks *Proc. 14th ACM SIGKDD Int. Conf. Knowledge discovery and data mining* pp 462–70
[18] Palla G, Barabási A-L and Vicsek T 2007 Quantifying social group evolution *Nature* **446** 664–7
[19] Zheleva E, Sharara H and Getoor L 2009 Co-evolution of social and affiliation networks *Proc. 15th ACM SIGKDD Int. Conf. Knowledge discovery and data mining* pp 1007–16

[20] Amaral L A N, Buldyrev S V, Havlin S, Leschhorn H, Maass P, Salinger M A, Stanley H E and Stanley M H R 1997 Scaling behavior in economics: I. Empirical results for company growth *J. Phys.* I **7** 621–33

[21] Stanley M H R, Amaral L A N, Buldyrev S V, Havlin S, Leschhorn H, Maass P, Salinger M A and Stanley H E 1996 Scaling behaviour in the growth of companies *Nature* **379** 804–6

[22] González-Val R 2019 Lognormal city size distribution and distance *Econ. Lett.* **181** 7–10

[23] Fazio G and Modica M 2015 Pareto or log-normal? Best fit and truncation in the distribution of all cities *J. Regional Sci.* **55** 736–56

[24] Zhu K, Li W, Fu X and Nagler J 2014 How do online social networks grow? *PLoS One* **9** e100023

[25] Kairam S R, Wang D J and Leskovec J 2012 The life and death of online groups: predicting group growth and longevity *Proc. 5th ACM Int. Conf. Web Search and Data Mining* pp 673–82

[26] Alstott J, Bullmore E and Plenz D 2014 Powerlaw: a python package for analysis of heavy-tailed distributions *PLoS One* **9** 1–11

[27] Mitzenmacher M 2004 A brief history of generative models for power law and lognormal distributions *Internet Math.* **1** 226–51

[28] Mondani H, Holme P and Liljeros F 2014 Fat-tailed fluctuations in the size of organizations: the role of social influence *PLoS One* **9** e100527

[29] Fu D, Pammolli F, Buldyrev S V, Riccaboni M, Matia K, Yamasaki K and Stanley H E 2005 The growth of business firms: theoretical framework and empirical evidence *Proc. Natl Acad. Sci. USA* **102** 18801–6

[30] Frasco G F, Sun J, Rozenfeld H D and Ben-Avraham D 2014 Spatially distributed social complex networks *Phys. Rev.* X **4** 011008

[31] Qian J-H, Chen Q, Han D-D, Ma Y-G and Shen W-Q 2014 Origin of Gibrat law in internet: asymmetric distribution of the correlation *Phys. Rev.* E **89** 062808

[32] Mitrović M, Paltoglou G and Tadić B 2011 Quantitative analysis of bloggers' collective behavior powered by emotions *J. Stat. Mech.* P02005

[33] Dankulov M M, Melnik R and Tadić B 2015 The dynamics of meaningful social interactions and the emergence of collective knowledge *Sci. Rep.* **5** 1–10

[34] Vranić A and Dankulov M M 2021 Growth signals determine the topology of evolving networks *J. Stat. Mech.* **2021** 013405

[35] Briët J and Harremoës P 2009 Properties of classical and quantum Jensen–Shannon divergence *Phys. Rev.* A **79** 052311

[36] Mansfield E 1962 Entry, Gibrat's law, innovation, and the growth of firms *Am. Econ. Rev.* **52** 1023–51

[37] Barthelemy M 2019 The statistical physics of cities *Nat. Rev. Phys.* **1** 406–15

[38] Barabási A-L, Albert R and Jeong H 1999 Mean-field theory for scale-free random networks *Physica* A **272** 173–87

https://doi.org/10.1088/1742-5468/aca0e9

Similarity-based Link Prediction from Modular Compression of Network Flows

Christopher Blöcker
Integrated Science Lab, Department of Physics
Umeå University
christopher.blocker@umu.se

Jelena Smiljanić[*]
Integrated Science Lab, Department of Physics
Umeå University
jelena.smiljanic@umu.se

Ingo Scholtes
Center for Artificial Intelligence and Data Science
University of Würzburg
ingo.scholtes@uni-wuerzburg.de

Martin Rosvall
Integrated Science Lab, Department of Physics
Umeå University
martin.rosvall@umu.se

Abstract

Node similarity scores are a foundation for machine learning in graphs for clustering, node classification, anomaly detection, and link prediction with applications in biological systems, information networks, and recommender systems. Recent works on link prediction use vector space embeddings to calculate node similarities in undirected networks with good performance. Still, they have several disadvantages: limited interpretability, need for hyperparameter tuning, manual model fitting through dimensionality reduction, and poor performance from *symmetric* similarities in *directed* link prediction. We propose *MapSim*, an information-theoretic measure to assess node similarities based on modular compression of network flows. Unlike vector space embeddings, *MapSim* represents nodes in a discrete, non-metric space of communities and yields *asymmetric* similarities in an unsupervised fashion. We compare *MapSim* on a link prediction task to popular embedding-based algorithms across 47 networks and find that *MapSim*'s average performance across all networks is more than 7% higher than its closest competitor, outperforming all embedding methods in 11 of the 47 networks. Our method demonstrates the potential of compression-based approaches in graph representation learning, with promising applications in other graph learning tasks.

1 Introduction

Calculating similarity scores between objects is a fundamental problem in machine learning tasks, from clustering, anomaly detection, and text mining to classification and recommender systems. In Euclidean feature spaces, similarities between feature vectors are commonly calculated as lengths, norms, angles, or other geometric concepts, possibly using *kernel functions* that perform implicit non-linear mappings to high-dimensional feature spaces [1]. For relational data represented as graphs, methods using the graph topology to calculate pairwise node similarities can address learning problems such as graph clustering, node classification, and link prediction. For link prediction, recent works take a multi-step approach and separate *representation learning* and *link prediction* [2, 3]: First, they learn a latent-space node embedding from the graph's topology, using methods such as graph or matrix factorisation [4, 5], or random walk-based techniques [6–8]. Then, they interpret node positions as points in a high-dimensional feature space, possibly applying downstream dimensionality reduction. Finally, they use node positions in the resulting feature space to assign new "features" to pairs of nodes, which can be used to predict links. Taking an unsupervised approach, links are predicted based on node similarities [9] by calculating distance metrics or similarity scores between

[*] Also with the Center for the Study of Complex Systems, Institute of Physics, University of Belgrade.

C. Blöcker et al., Similarity-based Link Prediction from Modular Compression of Network Flows. *Proceedings of the First Learning on Graphs Conference (LoG 2022)*, PMLR 198, Virtual Event, December 9–12, 2022.

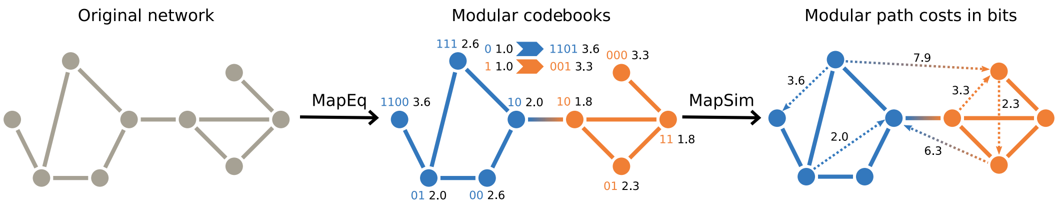

Figure 1: We calculate node similarities for predicting links based on a network's modular coding scheme of the map equation. Blue and orange nodes have a unique codeword within their module, shown next to the nodes and derived from their stationary visit rates. Decimal numbers show the theoretical lower limit for the codeword length in bits. *Map equation similarity*, *MapSim* for short, derives description lengths for predicted links, connecting more similar nodes uses fewer bits. Intra-community links tend to have shorter description lengths than inter-community links.

node pairs to rank them. We can alternatively use a supervised approach [10] by (i) using binary operators like the Hadamard product [7], (ii) sampling negative instances (node pairs not connected by links), and (iii) using the features of positive and negative instances to train a supervised binary classifier [7].

Advances in graph embedding and representation learning have considerably improved our ability to predict links in networks, with applications in biological [11] and social [12] networks and in recommender systems [13]. However, these methods introduce challenges for real-world link-prediction tasks: First, they require specifying hyperparameters that control aspects regarding the *scale* of patterns in graphs, the influence of local and non-local structures, and the latent space dimensionality [14]. Network-specific hyperparameter tuning addresses these issues, but is challenging in real applications and aggravates the risk of overfitting; recent systematic comparisons reveal that the performance of different methods largely varies across data sets [2, 3]. These challenges make it difficult for practitioners to choose and optimally parametrise an embedding method. Second, using latent metric spaces implies *symmetric* similarities, limiting the performance when predicting *directed* links [5, 15]. Third, compared with hand-crafted features, embeddings tend to have low interpretability: We can assess the similarity of nodes, but we cannot explain *why* some nodes are more similar than others [2–4]. Nevertheless, recent graph neural network-based approaches focus on learning features for link prediction from local subgraphs [16], overlapping node neighbourhoods [17], or shortest paths [18], achieving favourable performance. Finally, recent works highlight fundamental limitations of low-dimensional representations of complex networks [19], questioning to what extent Euclidean embeddings can capture patterns relevant to link prediction.

Motivated by recent works highlighting the importance of community structures for link prediction [2, 20, 21], we propose a novel approach to similarity-based link prediction that addresses these issues. Our contributions are:

- We introduce map equation similarity, *MapSim* for short, an information-theoretic method to calculate asymmetric node similarities. *MapSim* builds on the map equation [22], a framework that applies coding theory to compress random walks based on hierarchical cluster structures.
- Unlike other random walk-based embedding techniques, our work builds on an analytical approach to calculate the minimal expected description length of random walks, neither requiring simulating random walks nor tuning hyperparameters.
- Following the minimum description length principle, *MapSim* incorporates Occam's razor and balances explanatory power with model complexity, making dimensionality reduction superfluous. With hierarchical cluster structures, *MapSim* captures patterns at multiple scales simultaneously and combines the advantages of local and non-local similarity scores.
- We validate *MapSim* in an unsupervised, similarity-based link prediction task and compare its performance to six well-known embedding-based techniques in 47 empirical networks from different domains. This analysis highlights challenges in the generalisability of embedding techniques and parametrisations across different networks.
- Confirming recent surveys, we find that the performance of popular embedding techniques for unsupervised link prediction without network-specific hyperparameter tuning depends on the data. In contrast, *MapSim* provides high performance across a wide range of networks,

with an average performance 7.7% and 7.5% better than the best competitor in undirected and directed networks, respectively. *MapSim* outperforms the chosen baseline methods in 11 of the 47 networks with a worst-case performance 44% and 33% better than popular embedding techniques in undirected and directed networks, respectively.

In summary, we take a novel perspective on graph representation learning that fundamentally differs from other random walk-based graph embeddings. Instead of embedding nodes into a metric space, leading to symmetric similarities, we develop an unsupervised learning framework where (i) positions of nodes in a coding tree capture their representation in a non-metric latent space, and (ii) node similarities are calculated based on how well transitions between nodes are compressed by a network's hierarchical modular structure (figure 1). Apart from node similarities that can be "explained" based on community structures captured in the coding tree, *MapSim* yields asymmetric similarity scores that naturally support link prediction in directed networks. We provide a simple, non-parametric, and scalable unsupervised method with high generalisability across data sets. Our work demonstrates the power of compression-based approaches to graph representation learning, with promising applications in other graph learning tasks.

2 Related Work and Background

We first summarise recent works on graph embedding and similarity-based link prediction. Then, we review the map equation, an information-theoretic objective function for community detection and the theoretical foundation of our compression-based similarity score.

2.1 Related Work

Focusing on unsupervised similarity-based link prediction, we consider methods that calculate a bivariate function $\text{sim}(u, v) \in \mathbb{R}^d$, where $u, v \in V$ are nodes in a directed or undirected, possibly weighted graph $G = (V, E)$ [23, 24]. While similarity metrics often consider scalar functions ($d = 1$), recent vector space embeddings use binary operators to assign vector-valued "features" with $d > 1$ to node pairs. Since vectorial features are typically used in downstream classification techniques, this can be seen as an *implicit* mapping to similarities, for example "similar" features being assigned similar class probabilities. We limit our discussion to *topological or structural approaches* [23], and consider functions $\text{sim}(u, v)$ that can be calculated solely based on the edges E in graph G without requiring additional information such as node attributes or other non-topological graph properties.

Several works define scalar similarities based on local topological characteristics such as the Jaccard index of neighbour sets, degrees of nodes, or degree-weighted measures of common neighbours [25]. Other methods define similarities based on random walks, paths, or topological distance between nodes [9, 26–28]. Compared to purely local approaches, an advantage of random walk-based methods is their ability to incorporate both local and non-local information, which is crucial for sparse networks where nodes may lack common neighbours. Since walk-based methods reveal cluster patterns in networks [22], they generally perform well in downstream tasks such as link prediction and graph clustering [2]. Graph factorisation approaches that use eigenvectors of different types of *Laplacian matrices* that represent relationships between nodes share this high performance [29], likely because (i) Laplacians capture the dynamics of continuous-time random walks [30], and (ii) spectral methods can capture *small cuts* in graphs [31].

Building on these ideas, recent works on *graph representation learning* combine random walks and deep learning to obtain high-dimensional vector space embeddings of nodes, serving as features in downstream learning tasks [3, 14]: Perozzi et al. [6] generate a large number of short random walks to learn latent space representations of nodes by applying a word embedding technique that considers node sequences as word sequences in a sentence. This corresponds to an implicit factorisation of a matrix whose entries capture the logarithm of the expected probabilities to walk between nodes in a given number of steps [32]. Following a similar walk-based approach, Grover and Leskovec [7] generate node sequences with a biased random walker whose exploration behaviour can be tuned by *search bias* parameters p and q. The resulting walk sequences are used as input for the word embedding algorithm `word2vec` [33], which embeds objects in a latent vector space with configurable dimensionality. Tang et al. [8] construct vector space embeddings of nodes that simultaneously preserve first- and second-order proximities between nodes. Similar to Adamic and Adar [25], second-order node proximities are defined based on common neighbours. Extending the random walk approach in [6], Perozzi et al. [34] learn embeddings from so-called walklets, random walks that skip some nodes, resulting in embeddings that capture structural features at multiple scales.

The abovementioned graph embedding methods compute a representation of nodes in a, compared to the number of nodes in the network, low-dimensional Euclidean space. A suitably defined metric for *similarity* or *distance* of nodes enables recovering the link topology with high fidelity [35], forming the basis for similarity-based link prediction. In contrast, Lichtenwalter et al. [10] argued for a new perspective that uses supervised classifiers based on (i) multi-dimensional features of node pairs, and (ii) an undersampling of negative instances to address inherent class imbalances in link prediction. Recent applications of graph embedding to link prediction have taken a similar supervised approach, for example using vector-valued binary operators to construct features for node pairs from node vectors [6, 7, 24]. Despite good performance, recent works have cast a more critical light on such applications of low-dimensional graph embeddings. Questioning the distinction between deep learning-based embeddings and graph factorisation techniques, Qiu et al. [4] show that popular embedding techniques can be understood as (approximate) factorisations of matrices that capture graph topology. Thus, low-dimensional embeddings can be viewed as a (lossy) compression of graphs, while link prediction or graph reconstruction can be viewed as the decompression step. Fitting this view, a recent study of the topological characteristics of networks' low-dimensional Euclidean representations has highlighted fundamental limitations of embeddings to capture complex structures found in real networks [19].

Techniques like node2vec, LINE, or DeepWalk have been reported to perform well for link prediction despite those limitations. However, recent surveys concur that finetuning their hyperparameters to the specific data set is required [2, 21, 36], which can be problematic in large data sets and increase the risk of overfitting. When used for link prediction, graph embedding methods are typically combined with dimensionality reduction and supervised classification algorithms, possibly using non-linear kernels. Comparative studies found that the performance of Euclidean graph embeddings for link prediction is connected to their ability to represent communities in graphs as clusters in the feature space [2], which, due to the non-linear nature of graph data [37], strongly depends on their topology. Using symmetric operators or distance measures in metric spaces limits their ability to predict *directed* links because the ground truth for (u, v) can differ from (v, u) [15].

These issues raise the general question whether we should use low-dimensional Euclidean embeddings for link prediction tasks. Recent works addressed some of those open questions, for example with hyperbolic or non-linear embeddings [20, 37], extensions of Euclidean embeddings for directed link prediction [15], or embeddings that explicitly account for community structures [21, 38, 39]. However, existing works still use hyperparameters, require separate dimensionality reduction or model selection to identify the optimal number of dimensions, fail to capture rich hierarchically nested community structures present in real-world networks [40], or do not integrate community detection with representation learning. Addressing all issues at once, we take a novel approach that treats graph representation learning as a compression problem: We use the map equation [22], an analytical information-theoretic approach to compress flows of random walks in directed or undirected, possibly weighted networks based on their modular structure. Unlike recent work by Ghasemian et al. [41] that predicts links based on how they influence the map equation's estimated codelength, requiring inefficient recalculations, we take advantage of the map equation's coding machinery without any computational overhead. The map equation's hierarchical coding tree with node assignments provides an embedding in a discrete, non-metric latent space of possibly hierarchical community labels with automatically optimised dimensionality using a minimum description length approach. Following the map equation's compression principles, we relate the similarity between nodes u and v to how efficiently we can compress the link (u, v) with respect to the network's modular structure. As an analytical approach, our method neither introduces hyperparameters nor needs to simulate random walks, and naturally yields asymmetric node similarities suitable to predict directed links.

2.2 Background: the map equation

The map equation is an information-theoretic objective function for community detection that, conceptually, models network flows with random walks [22]. To detect communities, the map equation compresses the random walks' per-step description length by searching for sets of nodes with long flow persistence: network areas where a random walker tends to stay for a longer time.

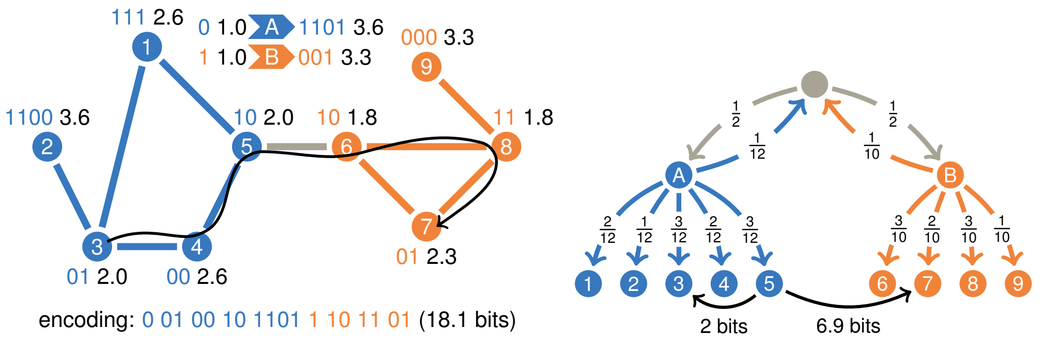

Figure 2: Map equation coding principles. **Left:** An example network with nine nodes, ten links, and two communities, A and B, indicated by colours. Each random-walker step is encoded by one codeword for intra-module transitions, or three codewords for inter-module transitions. Codewords are shown next to nodes in colours, their length in bits in the information-theoretic limit in black. Module entry and exit codewords are shown to the left and right of the coloured arrows, respectively. The black trace shows a possible section of a random walk with its encoding and theoretical length at the bottom. **Right:** The corresponding coding tree. Links are annotated with transition rates to calculate similarities in the information-theoretic limit. Each coding tree path corresponds to a network link, which may or may not exist. The coder remembers the random walker's module but not the most recently visited node. Describing the intra-module transition from node 5 to 3 requires $-\log_2(3/12) = 2$ bits. The inter-module transition from node 5 to 7 requires three steps and $-\log_2(1/12 \cdot 1/2 \cdot 2/10) \approx 6.9$ bits.

Consider a communication game where the sender observes a random walker on a network, and uses binary codewords to update the receiver about the random walker's location. In the simplest case, all nodes belong to the same module and we use a Huffman code to assign unique codewords to the nodes based on their stationary visit rates. With a one-module partition, M_1, the sender communicates one codeword per random-walker step to the receiver. The theoretical lower limit for the per-step description length, we call it *codelength*, is the entropy of the nodes' visit rates [42],

$$L(\mathsf{M}_1) = \mathcal{H}(P) = -\sum_{u \in V} p_u \log_2 p_u, \tag{1}$$

where $\mathcal{H}$ is the Shannon entropy, P is the set of the nodes' visit rates, and p_u is node u's visit rate.

In networks with modular structure, we can compress the random walks' description by grouping nodes into more than one module such that a random walker tends to remain within modules, and module switches become rare. This lets us re-use codewords across modules and design a codebook per module based on the nodes' module-normalised visit rates. However, sender and receiver need a way to encode module switches. The map equation uses a designated module-exit codeword per module and an index-level codebook with module-entry codewords. In a two-level partition, the sender communicates one codeword for intra-module random-walker steps to the receiver, or three codewords for inter-module steps (figure 2). The lower limit for the codelength is given by the sum of entropies associated with module and index codebooks, weighted by their usage rates. Given a partition of the network's nodes into modules, M, the map equation [22] formalises this relationship,

$$L(\mathsf{M}) = q\mathcal{H}(Q) + \sum_{\mathsf{m} \in \mathsf{M}} p_{\mathsf{m}} \mathcal{H}(P_{\mathsf{m}}). \tag{2}$$

Here $q = \sum_{\mathsf{m} \in \mathsf{M}} q_{\mathsf{m}}$ is the index-level codebook usage rate, q_{m} is the entry rate for module m, and $Q = \{q_{\mathsf{m}} \mid \mathsf{m} \in \mathsf{M}\}$ is the set of module entry rates; m_{exit} is the exit rate for module m, $p_{\mathsf{m}} = \mathsf{m}_{\text{exit}} + \sum_{u \in \mathsf{m}} p_u$ is the codebook usage rate for module m, and $P_{\mathsf{m}} = \{\mathsf{m}_{\text{exit}}\} \cup \{p_u \mid u \in \mathsf{m}\}$ is the set of node visit rates in m, including m's module exit rate.

The map equation can detect communities in simple, weighted, directed, and higher-order networks, and can be generalised to hierarchical partitions through recursion [40]. To make use of node metadata for detecting communities, we can either incorporate a corresponding term in the map equation [43],

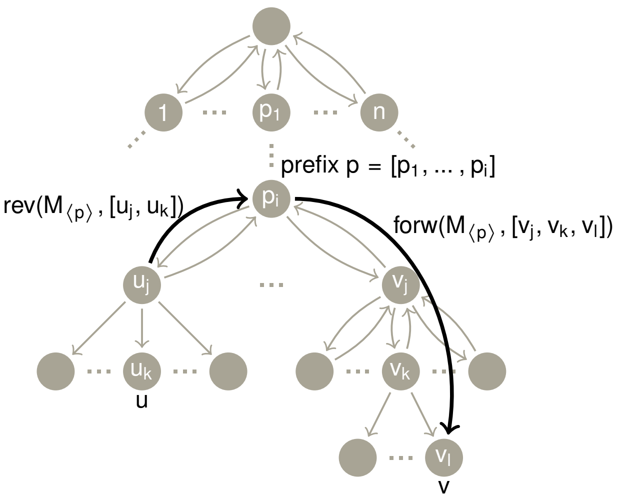

Figure 3: Illustration of map equation similarity between nodes u and v with addresses $\operatorname{addr}(\mathsf{M}, u) = [p_1, \ldots, p_i, u_j, u_k]$ and $\operatorname{addr}(\mathsf{M}, v) = [p_1, \ldots, p_i, v_j, v_k, v_l]$. M is the complete network partition. The longest common prefix between the addresses for u and v is $p = [p_1, \ldots, p_i]$, and $\mathsf{M}_{\langle p \rangle}$ is the sub-module at address p within M, that is the smallest module that contains u and v.

design metadata-informed flow models [44], or introduce a prior network and reinforce link weights between nodes with the same metadata label [45].

3 MapSim: node similarities from modular flow compression

Compression-based similarity measures consider pairs of objects more similar if they jointly compress better. Extending this idea to networks, we exploit the coding of network flows based on the map equation, and use it to calculate information-theoretic pairwise similarities between nodes: *MapSim*. We interpret a network's community structure as an implicit embedding and, roughly speaking, consider nodes in the same community as more similar than nodes in different communities.

To calculate node similarities, we begin with a network partition and its corresponding modular coding scheme[2], which can be visualised as a tree, annotated with the transition rates defined by the link patterns in the network (figure 2). While the network's topology constrains random walks to transitions along existing links, the coding scheme is more flexible and can describe transitions between *any* pair of nodes. To describe the transition from node u to v, we find the corresponding path in the partition tree and multiply the transition rates along that path, that is, we use the *coarse-grained description* of the network's community structure, not the network's actual link pattern; it can describe any transition regardless of whether the link (u, v) exists in the network or not. The description length in bits for a path with transition rate r is $-\log_2(r)$. For example, consider the scenario in figure 2 where we calculate similarity scores for the two directed links $(5, 3)$ and $(5, 7)$, neither of which exists in the network. Nodes 5 and 3 are in module A, and the rate at which a random walker in A visits node 3 is $3/12$, requiring $-\log_2(3/12) = 2$ bits to describe that transition. Node 7 is in module B, and a random walker in A exits A at rate $1/12$, enters B at rate $1/2$, and then visits node 7 at rate $2/10$, that is, at rate $1/120$, requiring $-\log_2(1/120) \approx 6.9$ bits.

Paths to derive similarities emanate from modules, not from nodes, because the model must generalise to unobserved data. If compression was our sole purpose, we would use node-specific codebooks containing codewords for neighbouring nodes, but no longer detect communities, and only be able to describe observed links. Instead, the map equation's coding scheme is designed to capitalise on modular network structures: The modular code structure provides a model that generalises to unobserved data, coarse-grains the path descriptions, and prevents overfitting.

For the general case, where M can be a hierarchical network partition, we number the sub-modules within each module m from 1 to n_{m} – we refer to these numbers as addresses – such that an ordered

[2] In principle, arbitrary network partitions can be used, regardless of the used community detection method.

sequence of addresses uniquely identifies a path starting at the root of the partition tree. We let $\mathrm{addr}\colon \mathsf{M} \times N \to \mathit{List}\,(\mathbb{N})$ be a function that takes a network partition and a node as input, and returns the node's address in the partition. To calculate the similarity of node v to u, we identify the longest common prefix p of the nodes' addresses, $\mathrm{addr}\,(\mathsf{M}, u)$ and $\mathrm{addr}\,(\mathsf{M}, v)$, and select the partition tree's sub-tree $\mathsf{M}_{\langle p \rangle}$ that corresponds to the prefix p: $\mathsf{M}_{\langle p \rangle}$ is the smallest sub-tree that contains u and v. We obtain the addresses for u and v within sub-tree $\mathsf{M}_{\langle p \rangle}$ by removing the prefix p from their addresses. That is, $\mathrm{addr}\,(\mathsf{M}, u) = p \mathbin{+\!\!\!+} \mathrm{addr}(\mathsf{M}_{\langle p \rangle}, u)$ and $\mathrm{addr}\,(\mathsf{M}, v) = p \mathbin{+\!\!\!+} \mathrm{addr}(\mathsf{M}_{\langle p \rangle}, v)$, where $\mathbin{+\!\!\!+}$ is list concatenation. The rate at which a random walker transitions from u to v is the product of (i) the rate at which the random walker moves along the path $\mathrm{addr}(\mathsf{M}_{\langle p \rangle}, u)$ in *reverse direction*, $\mathrm{rev}(\mathsf{M}_{\langle p \rangle}, \mathrm{addr}(\mathsf{M}_{\langle p \rangle}, u))$, that is from u to the root of $M_{\langle p \rangle}$, and (ii) the rate at which the random walker moves along the path $\mathrm{addr}(\mathsf{M}_{\langle p \rangle}, v)$ in *forward direction*, $\mathrm{forw}(\mathsf{M}_{\langle p \rangle}, \mathrm{addr}(\mathsf{M}_{\langle p \rangle}, v))$, that is from the root of $\mathsf{M}_{\langle p \rangle}$ to v, where

$$\mathrm{rev}\,(\mathsf{M}, a) = \begin{cases} 1 & \text{if } a = [x] \\ \mathsf{M}_{\langle [x] \rangle, \text{exit}} \cdot \mathrm{rev}(\mathsf{M}_{\langle [x] \rangle}, a') & \text{if } a = [x] \mathbin{+\!\!\!+} a' \end{cases} \tag{3}$$

$$\mathrm{forw}\,(\mathsf{M}, a) = \begin{cases} p_{\langle [x] \rangle} / p_{\mathsf{M}} & \text{if } a = [x] \\ \mathsf{M}_{\langle [x] \rangle, \text{enter}} \cdot \mathrm{forw}(\mathsf{M}_{\langle [x] \rangle}, a') & \text{if } a = [x] \mathbin{+\!\!\!+} a' \end{cases} \tag{4}$$

and a' denotes non-empty sequences. Here p_{M} is the codebook use rate for module M and $p_{\langle [x] \rangle}$ is the visit rate for the node identified by address x within the given module. The final addresses in equation 3 and equation 4 are treated differently, reflecting that the map equation forgets the most recently visited node.

We illustrate these ideas in a generic example (figure 3). In short, we define map equation similarity,

$$\mathrm{MapSim}\,(M, u, v) = \mathrm{rev}(\mathsf{M}_{\langle p \rangle}, \mathrm{addr}(\mathsf{M}_{\langle p \rangle}, u)) \cdot \mathrm{forw}(\mathsf{M}_{\langle p \rangle}, \mathrm{addr}(\mathsf{M}_{\langle p \rangle}, v)), \tag{5}$$

where p is the longest common prefix shared by the addresses of u and v in the partition tree defined by M. To express map equation similarity in terms of description length, we take the $-\log_2$ of MapSim and regard pairs of nodes that yield a shorter description length as more similar.

MapSim is asymmetric since module entry and exit rates are, in general, different and u and v can have different visit rates. MapSim is zero if one node is in a disconnected component; the exit rate for regions without out-links is zero, so the corresponding description length is infinitely long. This issue can be addressed with the regularised map equation [45], a Bayesian approach that introduces an empirical prior to model incomplete data with weak links between all pairs of nodes, where prior link strengths depend on the connection patterns of each node.

We calculate node similarities in three steps: (i) inferring a network's community with Infomap [46], a greedy, search-based optimisation algorithm for the map equation, (ii) representing the corresponding coding scheme in a suitable data structure, and (iii) using *MapSim* to computing similarities based on the coding scheme. The overall approach is illustrated in figs. 1 – 3 and algorithm 1.

Algorithm 1: Pseudo-code of function MapSim to calculate similarity score for node pair (u, v).

```
Input   : graph $G$ and pair of nodes $(u, v)$
Output  : similarity score of $(u, v)$
// Use Infomap to construct coding tree for compression
modules $=$ Infomap.minimiseMapEquation$(G)$
tree $=$ buildPartitionTree$(G,$ modules$)$
p $=$ longestCommonPrefix$($tree$, u, v)$
tree$_{\langle p \rangle}$ $=$ smallestSubtree$($tree, p$)$
// calculate code length of random walks from $u$ to $v$
addrU $=$ addr$($tree$_{\langle p \rangle}, u)$
addrV $=$ addr$($tree$_{\langle p \rangle}, v)$
revRate $=$ rev$($tree$_{\langle p \rangle}$, addrU$)$
fwdRate $=$ forw$($tree$_{\langle p \rangle}$, addrV$)$
return $-\log_2($revRate $\cdot$ fwdRate$)$
```

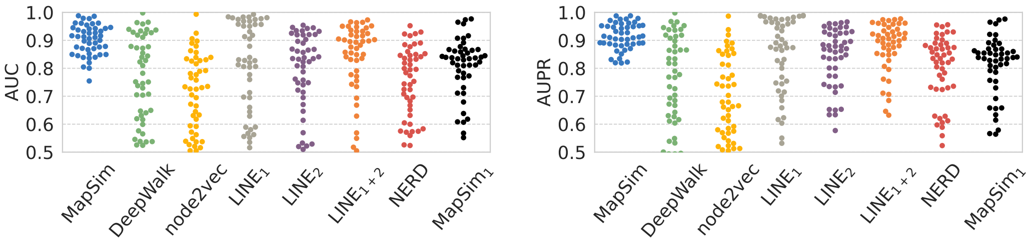

Figure 4: Link-prediction performance of MapSim, DeepWalk, node2vec, LINE, and NERD on 47 real-world networks. **Left:** AUC performance. **Right:** AUPR performance.

4 Experimental Validation

We evaluate the performance of *MapSim* in unsupervised, similarity-based link prediction for 47 real-world networks, 35 directed (table 1) and 12 undirected (table 3), retrieved from Netzschleuder [47] and Konect [48]. Details of the directed and undirected networks are shown in tables 2 and 4, respectively. Our analysis is based on a Python-implementation available on GitHub[3], building on Infomap, a fast and greedy search algorithm for minimising the map equation with an open source implementation in C++ [46, 49]. As baseline, we use four random walk and neighbourhood-based embedding methods: DeepWalk [6], node2vec [7], LINE [8], and NERD [15], using the respective author's implementation. We also include results for *MapSim* based on the one-module partition for each network for comparison, which ignores community structure. Adopting the argument by [7], we exclude graph factorisation methods and simple local similarity scores because they have already been shown to be inferior to node2vec. We include NERD because it is a recent random walk-based embedding method proposed for directed link prediction with higher reported performance than other walk-based embeddings [15].

4.1 Unsupervised Link Prediction

Different from works that use graph embeddings for *supervised* link prediction, we address *unsupervised* link prediction. Like Goyal and Ferrara [2] and Khosla et al. [15], we take a similarity-based approach that does not require training a classifier. We compute similarity scores based on node embeddings, rather than applying a supervised classifier to features computed for node pairs. We adopt the approach by Khosla et al. [15] and calculate node similarities as the sigmoid over the feature vectors' dot product.

Considering how different embedding techniques generalise across data sets, *we purposefully refrained from hyperparameter tuning*. We chose a single set of hyperparameters for each method, informed by the default parameters given by the respective authors and recent surveys' discussion regarding which hyperparameter values generally provide good link prediction performance. For DeepWalk and node2vec, we sample $r = 80$ random walks of length $l = 40$ per node, and use a window size of $w = 10$. For both methods, the underlying word embedding is applied using the default model parameters fixed by the authors, $skipgram = 1$, $k = 10$ and $mincount = 0$. For node2vec we set the return parameter to $p = 1$. Since for $q = p = 1$ node2vec is identical to DeepWalk, we use $q = 4$, which was found to provide good performance for link prediction [2]. We run LINE with first-order (LINE$_1$), second-order (LINE$_2$), and combined first-and-second-order proximity (LINE$_{1+2}$), use $1{,}000$ samples per node, and $s = 5$ negative samples. For NERD, we use 800 samples and $\kappa = 3$ negative samples per node. We set the number of neighbourhood nodes to $n = 1$, as suggested by the authors for link prediction. We use $d = 128$ dimensions for all embeddings. Since *MapSim* is a non-parametric method, it does not require setting any hyperparameters. However, to avoid local optima when heuristically minimising the map equation, we run Infomap 100 times and select the partition with the shortest description length.

We use 5-fold cross-validation to split links into train and test sets, treating weighted links as indivisible. We calculate the node embedding (for *MapSim* the coding tree) in the training network, derive predictions based on node similarities, and evaluate them based on the links in the validation set.

[3] https://github.com/mapequation/map-equation-similarity

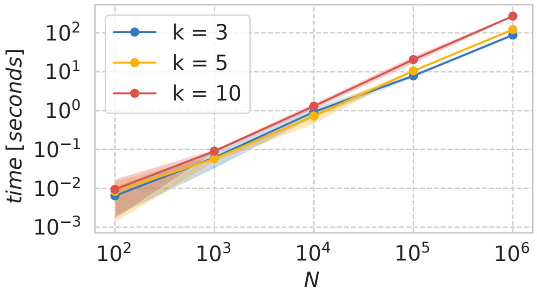

Figure 5: Runtime behavior for inferring the community structure with Infomap and constructing the coding tree for MapSim in synthetic k-regular networks with different size.

For each fold, we restrict the resulting training network to its largest (weakly) connected component. For a validation set with k positive links, we sample k negative links uniformly at random, and calculate scores for all $2k$ links. In undirected networks, for each positive link (u, v), we also consider (v, u) as positive, and, therefore, sample two negative links per positive link. Varying the discrimination threshold, we obtain a receiver operator characteristic (ROC) per fold, and calculate the area under the curve (AUC). Detailed results, including average and worst-case performance, are shown in tables 2 and 4; we also report precision-recall performance (table 5). We include *MapSim* based on the one-module partition[4] in the results and note that it performs better than using a modular partition in some cases: this suggests that the network does not have a strong community structure, which could be addressed with the regularised map equation [45]. When mentioning *MapSim* in the following, we refer to using modular partitions.

On average, *MapSim* outperforms all baseline methods across the 47 data sets in terms of AUC and AUPR (figure 4); for detailed results on a per-network basis see tables 2, 4, and 5 in the appendix. Using a one-sided two-sample t-test, we find that *MapSim*'s average performance across all networks is significantly higher than that of the best graph embedding method, LINE$_{1+2}$, both in directed and undirected networks ($p \approx 0.008$ and $p \approx 0.039$, respectively). *MapSim* provides the best performance in 11 of the 47 networks, with a standard deviation of the AUC score less than half of that of the best embedding-based method (LINE$_{1+2}$). For undirected networks, *MapSim* achieves the best performance for five of the 12 networks, while none of the embedding methods beats *MapSim's* performance in more than two networks. We find the largest performance gain in the directed network *linux*, where *MapSim* yields an increase of AUC of approximately 22.6% compared to the best embedding (NERD). *MapSim*'s worst-case performance across all networks is approximately 44% and 33% above that of the best-performing embedding for directed and undirected networks, respectively. *MapSim*' performance advantage can be as high as 84%, for example $AUC = 0.988$ of MapSim in *foursquare-friendships-new* vs. $AUC = 0.537$ for node2vec. While node2vec performs best in the largest directed network, *MapSim* performs best in the largest undirected network and in several small networks, suggesting that *MapSim* works well both for small and large networks.

We attribute those encouraging results to multiple features of our method: Different from graph embedding techniques that require downstream dimensionality reduction, *MapSim*'s compression approach implicitly includes model selection and avoids overfitting. Moreover, the representation of nodes in the coding tree is integrated with the optimisation of hierarchical community structures in the network. Due to its non-parametric approach and the use of the analytical map equation, *MapSim* performs well in absence of tuning to the specific data set.

4.2 Scalability Analysis

We analyse *MapSim*'s scalability in synthetically generated networks with modular structure and tunable size and link density. We generate k-regular random graphs with N nodes and (mean) degree k. To avoid trivial configurations where a modular structure is absent, we create a network by first generating two k-regular random graphs with $\frac{N}{2}$ nodes each and "cross" two links, one from each of the two graphs, to obtain a single connected network with strong community structure. We then apply Infomap to (i) minimise the map equation and extract the network's modular structure, and

[4] With the one-module partition, MapSim becomes equivalent to preferential attachment.

(ii) construct the coding tree for calculating node similarities. We repeat this 10 times for random networks with different numbers of N nodes and degrees k. The average run times are reported in figure 5, which shows that, for sparse networks, the runtime of *MapSim* is linear in the size of the network. Edler et al. [49] report that the theoretical asymptotic bound of computational complexity for the optimisation of the map equation is in $\mathcal{O}(N log N)$, which is the same as for vector space embedding techniques like node2vec and DeepWalk[5]. Thus, *MapSim* does not entail higher computational complexity compared to popular graph embeddings. This makes it an interesting choice for practitioners looking for a simple and scalable method that works well in small, large, directed, and undirected networks.

5 Conclusion and Outlook

We propose *MapSim*, a novel information-theoretic approach to compute node similarities based on a modular compression of network flows. Different from vector space embeddings, *MapSim* represents nodes in a discrete, non-metric space of communities that yields *asymmetric* similarities suitable to predict links in *directed* and *undirected* networks. The results are highly interpretable because the network's modular structure explains the similarities. Using description length minimisation, *MapSim* naturally accounts for Occam's razor, which avoids overfitting and yields a parsimonious coding tree. Performing unsupervised link prediction, we compare *MapSim* to popular embedding-based algorithms on 47 data sets covering networks from a few hundred to hundreds of thousands of nodes and millions of edges. Our analysis shows that the average performance of *MapSim* is more than 7% higher than its closest competitor, outperforming all competing methods in 11 of the 47 networks. Taking a new perspective on graph representation learning, our work demonstrates the potential of compression-based methods with promising applications in other graph learning tasks. Moreover, recent generalisations of the map equation to temporal and higher-order networks [49] suggest that our method also applies to graphs with non-dyadic or time-stamped relationships.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Ingo Scholtes acknowledges financial support from the Swiss National Science Foundation through grant No. 176938. Martin Rosvall was supported by the Swedish Research Council, Grant No. 2016-00796.

References

[1] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, 36(3):1171–1220, 2008. 1

[2] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018. ISSN 0950-7051. 1, 2, 3, 4, 8, 10

[3] Nino Arsov and Georgina Mirceva. Network embedding: An overview. *arXiv preprint arXiv:1911.11726*, 2019. 1, 2, 3, 10

[4] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proc. of the Eleventh ACM Intl. Conf. on Web Search and Data Mining*, WSDM '18, page 459–467, New York, 2018. ACM. ISBN 9781450355810. 1, 2, 4

[5] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016. 1, 2

[6] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014. 1, 3, 4, 8

[7] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016. 2, 3, 4, 8

[5] [2, 3] report linear complexity, which we could not confirm in the literature.

[8] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015. 1, 3, 8

[9] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007. 1, 3

[10] Ryan N. Lichtenwalter, Jake T. Lussier, and Nitesh V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, page 243–252, New York, NY, USA, 2010. ACM. ISBN 9781450300551. 2, 4

[11] Zachary Stanfield, Mustafa Coşkun, and Mehmet Koyutürk. Drug response prediction as a link prediction problem. *Scientific reports*, 7(1):1–13, 2017. 2

[12] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011. 2

[13] Hsinchun Chen, Xin Li, and Zan Huang. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, pages 141–142. IEEE, 2005. 2

[14] Mengjia Xu. Understanding graph embedding methods and their applications. *SIAM Review*, 63(4):825–853, 2021. 2, 3

[15] Megha Khosla, Jurek Leonhardt, Wolfgang Nejdl, and Avishek Anand. Node representation learning for directed graphs. In *Machine Learning and Knowledge Discovery in Databases*, pages 395–411, Cham, 2020. Springer. ISBN 978-3-030-46150-8. 2, 4, 8

[16] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5171–5181, Red Hook, NY, USA, 2018. Curran Associates Inc. 2

[17] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J. Kim. Neo-GNNs: Neighborhood overlap-aware graph neural networks for link prediction. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2

[18] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29476–29490. Curran Associates, Inc., 2021. 2

[19] C. Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, 117(11):5631–5637, 2020. ISSN 0027-8424. 2, 4

[20] Ali Faqeeh, Saeed Osat, and Filippo Radicchi. Characterizing the analogy between hyperbolic embedding and community structure of complex networks. *Phys. Rev. Lett.*, 121:098301, Aug 2018. 2, 4

[21] Yi-Jiao Zhang, Kai-Cheng Yang, and Filippo Radicchi. Systematic comparison of graph embedding methods in practical tasks. *Phys. Rev. E*, 104, 2021. ISSN 2470-0053. 2, 4

[22] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4):1118–1123, 2008. 2, 3, 4, 5

[23] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150 – 1170, 2011. ISSN 0378-4371. 3

[24] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys*, 49(4), 2016. 3, 4

[25] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3): 211–230, 2003. 3

[26] Zhenjiang Lin, Michael R Lyu, and Irwin King. Pagesim: a novel link-based measure of web page similarity. In *Proceedings of the 15th international conference on World Wide Web*, pages 1019–1020, 2006. 3

[27] Weiping Liu and Linyuan Lü. Link prediction based on local random walk. *EPL (Europhysics Letters)*, 89(5):58007, mar 2010.

[28] Joydeep Chandra, Ingo Scholtes, Niloy Ganguly, and Frank Schweitzer. A tunable mechanism for identifying trusted nodes in large scale distributed networks. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 722–729. IEEE, 2012. 3

[29] Matthew Brand and Kun Huang. A unifying theorem for spectral embedding and clustering. In *International Workshop on Artificial Intelligence and Statistics*, pages 41–48. PMLR, 2003. 3

[30] Naoki Masuda, Mason A. Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics Reports*, 716-717:1–58, 2017. ISSN 0370-1573. 3

[31] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001. 3

[32] Cheng Yang and Zhiyuan Liu. Comprehend deepwalk as matrix factorization. *arXiv preprint arXiv:1501.00358*, 2015. 3

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 3

[34] Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. Don't walk, skip! online learning of multi-scale network embeddings. In *Proc. of the 2017 IEEE/ACM Intl. Conf. on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, page 258–265, New York, NY, USA, 2017. ACM. ISBN 9781450349932. 4

[35] Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos E. Tsourakakis. Node embeddings and exact low-rank representations of complex networks, 2020. 4

[36] Thanh Tam Nguyen and Chi Thang Duong. A comparison of network embedding approaches. Technical report, School of Computer and Communication Sciences, EPFL, Technical Report, 2018. 4

[37] Xin Sun, Zenghui Song, Yongbo Yu, Junyu Dong, Claudia Plant, and Christian Böhm. Network embedding via deep prediction model. *arXiv preprint arXiv:2104.13323*, 2021. 4

[38] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 203–209. AAAI Press, 2017. 4

[39] Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 377–386, New York, NY, USA, 2017. ACM. ISBN 9781450349185. 4

[40] Martin Rosvall and Carl T Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one*, 6(4):e18209, 2011. 4, 5

[41] Amir Ghasemian, Homa Hosseinmardi, and Aaron Clauset. Evaluating overfit and underfit in models of network community structure. *IEEE Transactions on Knowledge and Data Engineering*, 32(9):1722–1735, 2020. 4

[42] Claude Elwood Shannon. A mathematical theory of communication. *Bell Labs Tech. J.*, 27(3): 379–423, 7 1948. 5

[43] Scott Emmons and Peter J. Mucha. Map equation with metadata: Varying the role of attributes in community detection. *Phys. Rev. E*, 100:022301, Aug 2019. doi: 10.1103/PhysRevE.100.022301. 5

[44] Aleix Bassolas, Anton Eriksson, Antoine Marot, Martin Rosvall, and Vincenzo Nicosia. Metadata-informed community detection with lazy encoding using absorbing random walks. *arXiv preprint arXiv:2111.05158*, 2021. 6

[45] Jelena Smiljanić, Christopher Blöcker, Daniel Edler, and Martin Rosvall. Mapping flows on weighted and directed networks with incomplete observations. *Journal of Complex Networks*, 9 (6), 12 2021. ISSN 2051-1329. 6, 7, 9

[46] D. Edler, A. Eriksson, and M. Rosvall. The infomap software package. https://www.mapequation.org, 2020. 7, 8

[47] Tiago P. Peixoto. The netzschleuder network catalogue and repository. https://networks.skewed.de/, 2020. 8

[48] Jérôme Kunegis. Konect: The koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, page 1343–1350, New York, NY, USA, 2013. ACM. ISBN 9781450320382. 8, 15, 17

[49] Daniel Edler, Ludvig Bohlin, and Martin Rosvall. Mapping Higher-Order Network Flows in Memory and Multilayer Networks with Infomap. *Algorithms*, 10:112, 2017. 8, 10

[50] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68:065103, Dec 2003. 15

[51] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, New York, 2005. ACM. ISBN 1595932151. 15

[52] Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H Brembeck, Heike Goehler, Martin Stroedicke, Martina Zenkner, Anke Schoenherr, Susanne Koeppen, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122 (6):957–968, 2005. 15

[53] Rob M Ewing, Peter Chu, Fred Elisma, Hongyan Li, Paul Taylor, Shane Climie, Linda McBroom-Cerajewski, Mark D Robinson, Liam O'Connor, Michael Li, et al. Large-scale mapping of human protein–protein interactions by mass spectrometry. *Molecular systems biology*, 3(1):89, 2007. 15

[54] Bureau of Transportation Statistics. T-100 domestic market. https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=310, 2017. 15

[55] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303 (5663):1538–1542, 2004. 15

[56] The openflights.org website. https://openflights.org/data.html, 2022. 15

[57] Paolo Massa, Martino Salvetti, and Danilo Tomasoni. Bowling alone and trust decline in social network sites. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 658–663, 2009. 15

[58] Michael Ley. The dblp computer science bibliography: Evolution, research issues, perspectives. In *String Processing and Information Retrieval*, pages 1–10, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45735-0. 15

[59] Michael Fire, Rami Puzis, and Yuval Elovici. *Link Prediction in Highly Fractional Data Sets*, pages 283–300. Springer New York, New York, NY, 2013. ISBN 978-1-4614-5311-6. 15

[60] B. Stabler. Transportation network test problems. https://github.com/bstabler/TransportationNetworks, 2022. 15

[61] M. Zaversnik V. Batagelj, A. Orvar. Network analysis of texts. *Language Technologies*, pages 143–148, 2002. 15

[62] Gergely Palla, Illés J Farkas, Péter Pollner, Imre Derényi, and Tamás Vicsek. Directed network modules. *New Journal of Physics*, 9(6):186–186, jun 2007. 15

[63] George R Kiss, Christine Armstrong, Robert Milroy, and James Piper. An associative thesaurus of english and its computer analysis. *The computer and literary studies*, pages 153–165, 1973. 15

[64] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000. 15

[65] Johannes Gehrke, Paul Ginsparg, and Jon Kleinberg. Overview of the 2003 kdd cup. *SIGKDD Explor. Newsl.*, 5(2):149–151, dec 2003. ISSN 1931-0145. 15

[66] Munmun De Choudhury, Hari Sundaram, Ajita John, and Dorée Duncan Seligmann. Social synchrony: Predicting mimicry of user actions in online social media. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 151–158, 2009. 15

[67] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, pages 217–226, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30115-8. 15

[68] Furkan Gursoy and Dilek Gunnec. Influence maximization in social networks under deterministic linear threshold model. *Knowledge-Based Systems*, 161:111–123, 2018. ISSN 0950-7051. 15

[69] Oliver Richters and Tiago P Peixoto. Trust transitivity in social networks. *PloS one*, 6(4): e18384, 2011. 15

[70] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM Workshop on Online Social Networks*, page 37–42, New York, 2009. ACM. ISBN 9781605584454. 15

[71] Vicenç Gómez, Andreas Kaltenbrunner, and Vicente López. Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, page 645–654, New York, NY, USA, 2008. ACM. ISBN 9781605580852. 15

[72] Kevin Gullikson. Python dependency analysis. `http://kgullikson88.github.io/blog/pypi-analysis.html`, 2016. 15

[73] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *The Semantic Web - ISWC 2003*, pages 351–368, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39718-2. 15

[74] Michael Fire, Lena Tenenboim-Chekina, Rami Puzis, Ofrit Lesser, Lior Rokach, and Yuval Elovici. Computationally efficient link prediction in a variety of social networks. *ACM Trans. Intell. Syst. Technol.*, 5(1), jan 2014. ISSN 2157-6904. 15, 17

[75] Manlio De Domenico, Antonio Lima, Paul Mougel, and Mirco Musolesi. The anatomy of a scientific rumor. *Scientific reports*, 3(1):1–9, 2013. 15

[76] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5–es, may 2007. ISSN 1559-1131. 15

[77] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Diameter of the world-wide web. *nature*, 401(6749):130–131, 1999. 15

[78] Munmun De Choudhury. Discovery of information disseminators and receptors on online social media. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, page 279–280, New York, 2010. ACM. ISBN 9781450300414. 15

[79] Samin Aref, David Friggens, and Shaun Hendy. Analysing scientific collaborations of new zealand institutions using scopus bibliometric data. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '18, New York, NY, USA, 2018. ACM. ISBN 9781450354363. 17

[80] Paolo Crucitti, Vito Latora, and Sergio Porta. Centrality measures in spatial networks of urban streets. *Phys. Rev. E*, 73:036125, Mar 2006. 17

[81] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998. 17

[82] Michael Fire, Rami Puzis, and Yuval Elovici. Organization mining using online social networks. *CoRR*, abs/1303.3741, 2013. 17

[83] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G.R. Gopinath, G.R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(suppl_1):D428–D432, 01 2005. ISSN 0305-1048. 17

[84] Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X*, 5:011027, Mar 2015. 17

[85] R. Alberich, J. Miro-Julia, and F. Rossello. Marvel universe looks almost like a real social network, 2002. 17

[86] Brian Karrer, M. E. J. Newman, and Lenka Zdeborová. Percolation on sparse networks. *Phys. Rev. Lett.*, 113:208702, Nov 2014. 17

[87] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170–180, 2015. ISSN 1084-8045. 17

A Appendix

Table 1: Properties of 35 directed networks, where weighted networks are marked with W, temporal link counts before aggregation into a static network are marked with $*$, and ρ is link reciprocity.

Data	Ref	Nodes	Edges	ρ
uni-email	[50]	1,133	10,903	1.000
polblogs	[51]	1,490	19,090	0.243
interactome-stelzl	[52]	1,706	6,207	0.972
interactome-figeys	[53]	2,239	6,452	0.006
us-air-traffic^{W}	[54]	2,278	*6,390,340	0.757
word-adjacency-japanese	[55]	2,704	8,300	0.073
openflights^{W}	[56]	3,214	66,771	0.978
jdk	[48]	6,434	150,985	0.009
advogato^{W}	[57]	6,541	51,127	0.307
word-adjacency-spanish	[55]	11,586	45,129	0.091
dblp-cite	[58]	12,590	49,759	0.004
anybeat	[59]	12,645	67,053	0.535
chicago-road	[60]	12,982	39,018	0.943
foldoc^{W}	[61]	13,356	120,238	0.479
google	[62]	15,763	171,206	0.254
word-assoc^{W}	[63]	23,132	312,342	0.094
cora	[64]	23,166	91,500	0.051
arxiv-citation-HepTh	[65]	27,770	352,807	0.003
digg-reply^{W}	[66]	30,398	*87,627	0.002
linux	[48]	30,837	213,954	0.002
arxiv-citation-HepPh	[65]	34,546	421,578	0.003
email-enron	[67]	36,692	367,662	1.000
inploid	[68]	39,749	57,276	0.272
pgp-strong	[69]	39,796	301,498	0.660
facebook-wall^{W}	[70]	46,952	*876,993	0.588
slashdot-threads^{W}	[71]	51,083	*140,778	0.210
python-dependency	[72]	58,743	108,399	0.004
lkml-reply^{W}	[48]	63,399	*1,096,440	0.635
epinions-trust	[73]	75,888	508,837	0.405
prosper	[48]	89,269	3,394,979	< 0.001
google-plus	[74]	211,187	1,506,896	0.482
twitter-higgs-retweet^{W}	[75]	256,491	328,132	0.005
amazon-copurchases-302	[76]	262,111	1,234,877	0.543
notre-dame-web	[77]	325,729	1,497,134	0.507
twitter-followers	[78]	465,017	834,797	0.003

Table 2: ROC AUC for link prediction in 35 directed networks for DeepWalk (DW), node2vec (n2v), LINE$_1$ (L$_1$), LINE$_2$ (L$_2$), LINE$_{1+2}$ (L$_{1+2}$), NERD, *MapSim* based on the one-level partition (MapSim$_1$), and *MapSim* based on modular partitions. Networks marked with W are weighted. † marks cases with AUC < 0.5 where we flipped the predicted link scores for AUC > 0.5. The best results per network are shown in bold, second-best underlined, and then rounded.

Data	DW	n2v	L$_1$	L$_2$	L$_{1+2}$	NERD	MS$_1$	MS
uni-email	0.911	†0.505	**0.957**	0.903	$\underline{0.932}$	0.667	0.711	0.852
polblogs	0.705	0.695	0.804	0.823	0.841	0.652	$\underline{0.868}$	**0.914**
interactome-stelzl	0.810	†0.505	**0.913**	0.758	$\underline{0.849}$	0.524	0.710	0.755
interactome-figeys	0.524	†0.828	†**0.905**	0.529	†$\underline{0.850}$	0.605	0.773	0.839
us-air-traffic^{W}	0.649	0.572	0.563	**0.935**	$\underline{0.933}$	0.774	0.858	0.916
word-adjacency-japanese	†0.538	†0.645	†0.580	0.748	0.743	0.526	**0.811**	$\underline{0.800}$
openflights^{W}	0.782	†0.665	0.918	0.934	**0.948**	0.708	0.838	$\underline{0.941}$
jdk	0.746	0.857	0.820	0.695	0.755	0.725	$\underline{0.974}$	**0.986**
advogato^{W}	0.738	0.563	0.806	0.865	**0.883**	0.742	0.812	$\underline{0.878}$
word-adjacency-spanish	†0.538	0.672	†0.713	**0.824**	0.791	0.632	$\underline{0.811}$	0.805
dblp-cite	0.840	†0.537	†0.589	0.646	0.549	$\underline{0.877}$	0.823	**0.890**
anybeat	0.647	0.539	0.644	0.841	**0.857**	0.683	0.834	$\underline{0.850}$
chicago-road	**0.998**	0.816	$\underline{0.981}$	0.670	0.835	†0.583	†0.608	0.848
foldoc^{W}	$\underline{0.927}$	0.549	**0.951**	0.832	0.905	0.571	0.618	0.845
google	0.844	0.792	0.831	0.868	$\underline{0.896}$	0.697	0.867	**0.962**
word-assoc^{W}	0.729	0.830	0.813	0.869	**0.916**	$\underline{0.884}$	0.837	0.849
cora	$\underline{0.939}$	0.839	**0.950**	0.761	0.831	0.830	0.839	0.906
arxiv-citation-HepTh	0.878	0.839	**0.958**	0.857	0.901	0.850	0.842	$\underline{0.942}$
digg-reply^{W}	†0.546	0.618	†0.552	0.714	0.693	$\underline{0.841}$	**0.845**	0.836
linux	0.704	0.726	0.567	0.722	0.734	0.784	$\underline{0.959}$	**0.961**
arxiv-citation-HepPh	$\underline{0.959}$	0.897	**0.975**	0.835	0.898	0.860	0.830	0.942
email-enron	0.823	†0.594	**0.983**	0.946	$\underline{0.963}$	0.819	0.840	0.931
inploid	0.631	0.766	0.516	0.838	0.828	0.753	$\underline{0.845}$	**0.870**
pgp-strong	0.873	0.527	**0.984**	0.890	0.924	0.795	0.782	$\underline{0.925}$
facebook-wall^{W}	$\underline{0.877}$	0.789	**0.931**	0.809	0.855	0.813	0.768	0.867
slashdot-threads^{W}	0.565	0.781	0.629	0.748	0.771	0.796	**0.877**	$\underline{0.876}$
python-dependency	0.751	0.735	†0.556	0.520	†0.505	0.832	**0.965**	$\underline{0.913}$
lkml-reply^{W}	0.537	0.731	0.590	**0.945**	$\underline{0.944}$	0.724	0.908	0.933
epinions-trust	0.599	0.777	0.806	$\underline{0.943}$	**0.952**	0.887	0.916	0.937
prosper	0.828	0.631	0.697	†0.614	†0.518	**0.952**	0.891	$\underline{0.945}$
google-plus	0.752	0.725	**0.957**	0.787	0.893	0.891	0.862	$\underline{0.946}$
twitter-higgs-retweet^{W}	0.620	$\underline{0.879}$	†0.695	†0.522	†0.569	0.799	**0.977**	0.820
amazon-copurchases-302	$\underline{0.963}$	0.826	**0.980**	0.896	0.936	0.575	0.638	0.910
notre-dame-web	$\underline{0.965}$	0.926	**0.975**	0.919	0.964	0.923	0.867	0.962
twitter-followers	0.526	†**0.993**	†$\underline{0.993}$	0.510	†0.973	0.917	0.809	0.871
Average	0.750	0.719	0.802	0.786	$\underline{0.832}$	0.757	0.830	**0.892**
Worst	0.524	0.505	0.516	0.510	0.505	0.524	$\underline{0.608}$	**0.755**
Standard Deviation	0.148	0.131	0.164	0.129	0.128	0.118	$\underline{0.088}$	**0.054**

Table 3: Properties of 12 undirected networks, where weighted networks are marked with W.

Data	Ref	Nodes	Edges
new-zealand-collab^{W}	[79]	1,511	4,273
urban-streets-venice	[80]	1,840	2,407
urban-streets-ahmedabad	[80]	2,870	4,387
power	[81]	4,941	6,594
facebook-organizations-L1	[82]	5,793	45,266
reactome	[83]	6,327	147,547
physics-collab-arXiv^{W}	[84]	14,488	59,026
marvel-universe	[85]	19,428	95,497
internet-as	[86]	22,963	48,436
marker-cafe	[74]	69,413	1,644,849
livemocha	[48]	104,103	2,193,083
foursquare-friendships-new	[87]	114,324	607,333

Table 4: ROC AUC on 12 undirected networks for DeepWalk (DW), node2vec (n2v), LINE$_1$ (L$_1$), LINE$_2$ (L$_2$), LINE$_{1+2}$ (L$_{1+2}$), NERD, *MapSim* based on the one-level partition (MS$_1$), and *MapSim* based on modular partitions (MS). Networks marked with W are weighted. † marks cases with AUC < 0.5 where we flipped the predicted link scores for AUC > 0.5. The best results per network are shown in bold, second-best underlined, and then rounded.

Data	DW	n2v	L$_1$	L$_2$	L$_{1+2}$	NERD	MS$_1$	MS
new-zealand-collab^{W}	0.616	0.734	†0.660	**0.921**	0.895	†0.559	0.834	0.839
urban-streets-venice	0.872	0.834	0.777	0.570	0.668	0.573	†0.607	**0.889**
urban-streets-ahmedabad	**0.939**	0.890	0.828	†0.533	0.629	†0.575	†0.731	0.897
power	0.919	0.863	0.827	0.741	0.777	0.600	0.552	**0.959**
facebook-organizations-L1	0.937	0.516	0.968	0.954	0.966	0.846	0.864	**0.979**
reactome	0.934	0.592	**0.983**	0.925	0.950	0.846	0.820	0.978
physics-collab-arXiv^{W}	0.929	0.521	**0.977**	0.807	0.871	0.695	0.568	0.955
marvel-universe	0.854	†0.633	0.879	0.834	**0.902**	0.852	0.679	0.900
internet-as	0.641	†0.705	0.535	0.921	0.920	0.744	0.766	**0.927**
marker-cafe	0.576	0.906	0.760	0.920	0.914	**0.930**	0.907	0.916
livemocha	0.708	0.758	0.839	0.861	0.876	**0.924**	0.855	0.876
foursquare-friendships-new	0.924	0.537	0.968	0.932	0.950	0.836	0.791	**0.988**
Average	0.821	0.707	0.834	0.826	0.860	0.748	0.748	**0.925**
Worst	0.576	0.521	0.535	0.533	0.629	0.559	0.552	**0.839**
Standard Deviation	0.136	0.140	0.132	0.137	0.106	0.136	0.116	**0.045**

Table 5: Average precision on 47 directed and undirected networks for DeepWalk (DW), node2vec (n2v), LINE$_1$ (L$_1$), LINE$_2$ (L$_2$), LINE$_{1+2}$ (L$_{1+2}$), NERD, *MapSim* based on the one-level partition (MapSim$_1$), and *MapSim* based on modular partitions. Weighted networks are marked with W. Results marked with † correspond to cases with AUC < 0.5 where we flipped the predicted link scores. Results are rounded, the best results are shown in bold, second-best are underlined.

Data	DW	n2v	L$_1$	L$_2$	L$_{1+2}$	NERD	MS$_1$	MS
uni-email	0.914	†0.513	**0.964**	0.916	$\underline{0.940}$	0.736	0.692	0.870
polblogs	0.627	0.631	0.817	0.838	$\underline{0.853}$	0.724	0.851	**0.903**
new-zealand-collab^{W}	0.643	0.661	†0.768	**0.925**	$\underline{0.907}$	†0.606	0.855	0.865
interactome-stelzl	0.835	†0.513	**0.944**	0.773	$\underline{0.853}$	0.612	0.757	0.820
urban-streets-venice	**0.897**	0.870	0.828	0.634	0.711	0.597	†0.564	$\underline{0.890}$
interactome-figeys	0.533	†0.703	†**0.889**	0.653	$^\dagger\underline{0.865}$	0.730	0.730	0.819
us-air-traffic^{W}	0.616	0.552	0.685	**0.937**	$\underline{0.934}$	0.835	0.833	0.903
word-adjacency-japanese	†0.494	†0.570	†0.623	0.801	0.796	0.628	**0.855**	$\underline{0.831}$
urban-streets-ahmedabad	**0.953**	0.919	0.864	†0.577	0.685	†0.523	†0.658	$\underline{0.915}$
openflights^{W}	0.767	†0.621	0.934	$\underline{0.950}$	**0.960**	0.798	0.840	0.950
power	$\underline{0.936}$	0.897	0.874	0.800	0.828	0.620	0.566	**0.962**
facebook-organizations-L1	0.919	0.508	**0.977**	0.966	0.974	0.882	0.835	$\underline{0.976}$
reactome	0.908	0.580	**0.985**	0.944	0.961	0.890	0.786	$\underline{0.978}$
jdk	0.777	0.862	0.891	0.737	0.807	0.761	$\underline{0.973}$	**0.987**
advogato^{W}	0.769	0.505	0.868	$\underline{0.892}$	**0.905**	0.805	0.810	0.890
word-adjacency-spanish	†0.496	0.652	†0.754	**0.863**	0.848	0.732	$\underline{0.863}$	0.851
dblp-cite	0.834	†0.485	†0.551	0.742	0.646	**0.908**	0.828	$\underline{0.905}$
anybeat	0.672	0.523	0.748	$\underline{0.884}$	**0.894**	0.784	0.867	0.883
chicago-road	**0.998**	0.863	$\underline{0.986}$	0.735	0.874	†0.559	†0.579	0.909
foldoc^{W}	$\underline{0.946}$	0.575	**0.966**	0.848	0.914	0.629	0.658	0.888
physics-collab-arXiv^{W}	0.939	0.592	**0.983**	0.858	0.899	0.725	0.634	$\underline{0.964}$
google	0.859	0.775	0.903	0.878	$\underline{0.907}$	0.775	0.889	**0.976**
marvel-universe	0.864	†0.666	**0.914**	0.840	0.899	0.884	0.615	$\underline{0.910}$
internet-as	0.685	†0.742	0.659	0.930	$\underline{0.930}$	0.822	0.817	**0.932**
word-assoc^{W}	0.727	0.846	0.873	0.896	**0.922**	$\underline{0.902}$	0.848	0.862
cora	$\underline{0.938}$	0.815	**0.958**	0.834	0.880	0.847	0.826	0.926
arxiv-citation-HepTh	0.865	0.812	**0.966**	0.896	0.925	0.868	0.839	$\underline{0.952}$
digg-reply^{W}	†0.501	0.585	†0.604	0.772	0.761	**0.873**	$\underline{0.835}$	0.834
linux	0.734	0.663	0.701	0.733	0.754	0.835	$\underline{0.959}$	**0.965**
arxiv-citation-HepPh	0.952	0.890	**0.975**	0.881	0.923	0.870	0.813	$\underline{0.952}$
email-enron	0.816	†0.541	**0.988**	0.963	$\underline{0.974}$	0.873	0.860	0.949
inploid	0.667	0.736	0.532	$\underline{0.879}$	0.875	0.819	0.869	**0.891**
pgp-strong	0.879	0.568	**0.989**	0.927	0.946	0.848	0.804	$\underline{0.954}$
facebook-wall^{W}	0.865	0.744	**0.951**	0.865	$\underline{0.890}$	0.833	0.753	0.890
slashdot-threads^{W}	0.604	0.769	0.744	0.835	0.848	0.855	$\underline{0.883}$	**0.886**
python-dependency	0.790	0.763	†0.715	0.653	†0.632	0.889	**0.965**	$\underline{0.915}$
lkml-reply^{W}	0.494	0.612	0.719	**0.959**	$\underline{0.958}$	0.821	0.920	0.942
marker-cafe	0.539	0.853	0.832	$\underline{0.921}$	0.917	**0.949**	0.901	0.912
epinions-trust	0.611	0.679	0.875	$\underline{0.960}$	**0.964**	0.925	0.921	0.947
prosper	0.818	0.531	0.616	†0.650	†0.465	**0.956**	0.855	$\underline{0.927}$
livemocha	0.694	0.737	0.880	0.868	$\underline{0.884}$	**0.930**	0.854	0.881
foursquare-friendships-new	0.918	0.520	$\underline{0.976}$	0.948	0.961	0.858	0.792	**0.988**
google-plus	0.704	0.679	**0.960**	0.870	0.921	0.921	0.870	$\underline{0.960}$
twitter-higgs-retweet^{W}	0.630	$\underline{0.880}$	†0.800	†0.634	†0.707	0.874	**0.976**	0.822
amazon-copurchases-302	$\underline{0.966}$	0.850	**0.987**	0.931	0.957	0.589	0.656	0.946
notre-dame-web	0.967	0.938	**0.980**	0.946	0.971	0.930	0.891	$\underline{0.971}$
twitter-followers	0.549	$^\dagger\underline{0.987}$	†**0.989**	0.714	†0.977	0.955	0.839	0.887