Efficient Methods for Influence-Based Network-Oblivious Community Detection

NICOLA BARBIERI, Yahoo Labs, London, United Kingdom
FRANCESCO BONCHI, The ISI Foundation, Turin, Italy, and Eurecat, Barcelona, Spain
GIUSEPPE MANCO, ICAR-CNR, Rende, Italy

We study the problem of detecting social communities when the social graph is not available but instead we have access to a log of user activity, that is, a dataset of tuples (u, i, t) recording the fact that user u "adopted" item i at time t. We propose a stochastic framework that assumes that the adoption of items is governed by an underlying diffusion process over the unobserved social network and that such a diffusion model is based on *community-level influence*. That is, we aim at modeling communities through the lenses of *social contagion*. By fitting the model parameters to the user activity log, we learn the community membership and the level of influence of each user in each community. The general framework is instantiated with two different diffusion models, one with discrete time and one with continuous time, and we show that the computational complexity of both approaches is linear in the number of users and in the size of the propagation log. Experiments on synthetic data with planted community structure show that our methods outperform non-trivial baselines. The effectiveness of the proposed techniques is further validated on real-word data, on which our methods are able to detect high-quality communities.

CCS Concepts: • **Mathematics of computing** → **Probabilistic algorithms**; • **Information systems** → **Clustering**; *Social networking sites*; • **Computing methodologies** → **Topic modeling**; **Latent variable models**; **Machine learning algorithms;**

Additional Key Words and Phrases: Social influence, information diffusion, social network analysis, network-oblivious community detection

ACM Reference Format:
Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2016. Efficient methods for influence-based network-oblivious community detection. ACM Trans. Intell. Syst. Technol. 8, 2, Article 32 (December 2016), 31 pages.
DOI: http://dx.doi.org/10.1145/2979682

1. INTRODUCTION

Detecting close-knit communities of like-minded people in on-line social networks is an important mining task with plenty of applications. Knowing groups of users with similar interests and a short distance on the social graph allows the analyst to develop more personalized user experiences and thus better web and mobile applications. For companies advertising and selling products through the Internet, the community structure of the social network provides invaluable knowledge. If a user responded positively to a certain ad, campaign, or product offer, then one might want to target other users in the same community: (i) By *homophily*, one can expect similar users to be more interested in the same product than random users, and (ii) if more users in the same

Authors' addresses: N. Barbieri, Tumblr, 35 E 21st St, New York, NY 10010; email: nicola@tumblr.com; F. Bonchi, ISI Foundation, Via Alassio, 10126 Torino, Italy; email: francesco.bonchi@isi.it; G. Manco, ICAR-CNR, Via Bucci 41c, 87036 Rende (CS), Italy; email: manco@icar.cnr.it.

community adopt the same product, then this might eventually create a *word-of-mouth* buzz, triggering more adoptions in the same community.

While the literature on community detection is wide, the real-world applications deployed so far are rather limited, often regarding just simple data analysis for social sciences. This is due to a fundamental observation that has been largely overlooked in the literature: The real-world applications that would benefit more from knowing the community structure of a social network *actually do not have access to the network.*

It is a matter of fact that the social network platforms are owned by a third party, such as Facebook or Twitter, who have realized that their proprietary social graph is an asset of inestimable value.[1] Thus they keep it secret for the sake of commercial competitive advantage, as well as due to privacy legislation. Take Facebook as an example: Business owners can set up a Facebook Page and create display ads or promoted posts to reach users,[2] but they are not allowed[3] to reconstruct the social graph and thus implement targeted ads campaign based on the knowledge of the communities. Another example is Twitter, which sells its Firehose (the complete stream of tweets is approximately half a billion per day) to other companies[4]: Regardless, the business partnership of the social graph is not disclosed, and actually, its reconstruction for commercial purposes is *explicitly forbidden by contract*.

In this article, we tackle the ambitious problem of inferring the community structure when the social graph is not available and without attempting to reconstruct it. A first step towards this goal is to analyze the alternative dynamics and data that we can exploit. A company advertising or developing applications over an on-line social network owns the log of user activity that it produces. In general, we might think of the activity log $\mathbb{D}$ as a set of tuples (u, i, t) that records the timestamp at which the user u "acted on" or "adopted" the item i: For instance, user u bought song i, user u clicked on ad i, user u liked photo i.

The key idea at the basis of this work is to exploit the phenomenon of social contagion to detect communities by analyzing, *exclusively*, the activity log $\mathbb{D}$. The basic assumptions are that (i) information can spread only by exploiting the social connections among users and that (ii) the network has a community structure, where communities are densely connected internally and loosely connected with other communities. As a consequence of these two assumptions, social contagion acts mainly locally, inside each community. Thus, if we see a group of users acting on item i in a short time frame, and we observe this occurring on various different items, then we can infer that these users are connected in some social network and that they communicate and can influence each other.

The following analogy might be appropriate. We might think of the unobserved social network as an underground network of channels and caves that we want to map. By letting some fluid enter into the network and monitoring how it flows, we can understand where there are caves in which the fluid can flow easily and accumulate (social communities) and where instead there are channels that allow flow from one cave to another and how easily the flow moves through each channel.

It is worth noting that the unobserved social network is not necessarily unique and clearly defined: Users can communicate through different media, for example, e-mail, telephone, Facebook, Twitter, Skype, or WhatsApp (just to mention a few), or they can even communicate in the real world, for example, while drinking a beer together. This does not make any difference for our setting: We just observe the adoptions and their

[1]http://techcrunch.com/2013/01/24/my-precious-social-graph/.
[2]https://www.facebook.com/business.
[3]https://developers.facebook.com/policy.
[4]http://gigaom.com/2012/11/29/as-the-firehose-matures-twitter-tightens-grip-on-valuable-asset/.

timing. Being network oblivious, our framework also solves the problem of *detecting communities over multiple networks*.

Finally, our method not only is aimed at detecting communities by exploiting social influence evidence: As a by-product, it also defines the level of influence of each user in each community. This allows us to identify for each community the "key" users, that is, the leaders who are most likely to influence the rest of the community to adopt a certain item. Those are the best users to target in a viral marketing campaign.

1.1. The Proposed Method

One possible approach to solve the network-oblivious community detection problem could be to first use the log $\mathbb{D}$ to infer the overall structure of the network and then apply some standard community detection techniques over the reconstructed social network. However, this approach has several drawbacks. First, methods for network reconstruction [Gomez Rodriguez et al. 2010, 2011] are inherently quadratic in the number of nodes, and the proposed inference procedure, which is based on convex optimization, is not easily scalable. Second, the reconstruction of the network might be explicitly forbidden by contract (as discussed above for Twitter and Facebook). Third, if the community detection is the ultimate goal then, as we show in our experiments in Section 6, it is more effective to go directly for it without passing through network reconstruction.

In this work, we propose a general framework for directly detecting communities in a network-oblivious setting without attempting to reconstruct the network. In particular, our proposal assumes that item adoptions are governed by an underlying stochastic diffusion process over the unobserved social network and that such a diffusion model is based on community-level influence. By fitting the model parameters to the user activity log $\mathbb{D}$, we learn the community membership and the influence level of each user in each community.

The general framework can be instantiated with different choices of diffusion models. In this article, we propose two models. First, we extend to the community level the classic (discrete-time) *Independent Cascade* model [Kempe et al. 2003]. The key idea is to assume that each user exerts the same degree of influence over a whole community. Then, we provide a finer-grained modeling by directly focusing on activation times. Here we assume that each user induces a fixed delay on the activation times of social peers within the same community.

1.2. Summary of Contributions and Roadmap

The main contributions of our work can be summarized as follows:

—We introduce the problem of network-oblivious community detection, exploiting user activity information. Despite the wide literature on the subjects of community detection and influence in social networks (briefly reviewed in Section 2), to the best of our knowledge we are the first ones to study community detection without the network.

—We define a stochastic framework for modeling users membership in communities based on community-level social influence. We devise an *expectation maximization* (EM) learning algorithm that embeds a penalized likelihood with negative Dirichlet-type prior. This enables a community annihilation mechanism, allowing the automatic detection of the best fitting number of communities (Section 3).

—We instantiate the general framework by considering two different diffusion models: a discrete-time community-level independent cascade model (Section 4) and a model based on the time delay between adoptions (Section 5). Notably, the computational complexity of both approaches is linear in the number of users and in the size of the propagation log. The inference phase is based on the above mentioned EM scheme for learning the model parameters. By exploiting the properties of the underlying propagation models, each iteration of the learning procedure can be efficiently computed in at most two scans of the propagation log.

—We run extensive experimentation on synthetic data with planted community structure (Section 6). Our results show that our methods outperform three non-trivial baselines. Experiments on a real-world Twitter dataset confirm the high quality of the discovered communities.

In Section 7 we conclude the article and discuss possible extensions. A six-page preliminary version of this work was presented in Barbieri et al. [2013b].

2. RELATED WORK

In this section, we briefly review related prior work. First, we discuss data-mining models in the area of social influence, and then we discuss a few articles at the intersection of social contagion and community detection.

2.1. Social Contagion

The term *social contagion* refers to the spread of new practices, beliefs, technologies, and products through a population, driven by *social influence*. It is a very central theme in social sciences, and recently it has attracted a lot of interest in the data-mining community [Bonchi 2011]. Fueled by the seminal work by Domingos and Richardson [2001] and Kempe et al. [2003], most of the attention has been devoted to exploiting social influence for "word-of-mouth" driven viral marketing applications.

Given a social network, where each arc (u, v) is associated with a weight (or probability) $p_{u,v}$ representing the strength of influence that u exerts over v, the problem is that of selecting the set of initial users that are more likely to influence the largest number of users in the social network, according to an assumed underlying *propagation model*. In this context, most of the effort has been devoted to develop efficient and scalable algorithms [Kimura and Saito 2006; Leskovec et al. 2007b; Chen et al. 2010; Goyal et al. 2011].

Other researchers have considered the social network and the log of past user activity jointly and studied important problems such as learning the parameters of the propagation model, that is, the strength of influence along each arc [Saito et al. 2008; Goyal et al. 2010] or how to distinguishing real social influence from "homophily" [Anagnostopoulos et al. 2008; Crandall et al. 2008; La Fond and Neville 2010]. Finally, a vast literature exists on the analysis of social influence in specific domains: for instance, studying person-to-person recommendation for purchasing books and videos [Leskovec et al. 2006, 2007a], telecommunications services [Shawndra et al. 2006], or studying information cascades driven by social influence in Twitter [Bakshy et al. 2011; Romero et al. 2011]. Weng et al. [2013] studies the role of information diffusion in the evolution of social networks: Their experimental results show that, as time progresses, the dynamics of information flow become an important component for the growth of the network.

More related to our work is the research by Gomez Rodriguez et al. [2010, 2011]. In this line of research, the social network is not given in input, and the problem is how to reconstruct the unobserved network starting from the log of users activity. The problem of *network reconstruction* is addressed by assuming that infections follow a continuous-time independent cascade model: Each node in a cascade is infected by at most one already-infected node, and thus a propagation is a directed tree. In NetInf [Gomez Rodriguez et al. 2010], the network is inferred by assuming that the probability of propagation between any pair of nodes is decreasing in the difference (always positive by assumption) of their infection times. In NetRate [Gomez Rodriguez et al. 2011], if the node u succeeds in activating v, then the contagion of the latter happens after an incubation time sampled from a chosen distribution, which defines the conditional likelihood of transmission between each pair of nodes and actually depends on the difference of their activation times.

According to this propagation model, the likelihood of a propagation cascade can be formulated by applying standard survival analysis [Lee and Wang 2003] in terms of survival (which models the probability that a node survives uninfected until a time T) and hazard fuctions (which models instantaneous infections). Authors study three different underlying distribution functions, namely the *exponential*, *Reileigh*, and *power-law*, and show that the likelihood of the observed data is convex and the trasmission rate parameters can hence be estimated by standard convex optimization procedures. In our experiments in Section 6, we use as a baseline NetRate to reconstruct the network, followed by Metis [Karypis and Kumar 1999] to partition the network.

Recent proposals have also focused on alternative ways of representing information diffusion interactions between nodes, mainly using latent-dimensional embedding techniques. Bourigault et al. [2014] proposes a framework based on a *heat diffusion process* that projects each node into a latent space where the proximity between a pair of nodes reflects the proximity of their activations times in the observed cascades. Similarly, Wang et al. [2015] introduces a factorization technique that associates two low-dimensional vectors to each node, representing influence and susceptibility, respectively.

2.2. Communities and Social Contagion

The study of social contagion is intrinsically connected to the problem of understanding the community structure of networks. In fact, individuals tend to adopt the behavior of their social peers, so social contagion happens first locally, within close-knit communities, and spreads virally only when it is able cross the boundaries of these densely connected clusters of people. Regardless of the vast literature on community detection algorithms (see Fortunato [2010] for a survey), there has not been much research at the intersection of community detection and social contagion.

Wang et al. [2010] studies the problem of finding the top-k influential users in *mobile* social networks. The authors propose to first detect communities and then assume that the influence of a user is limited to his or her community. Therefore they propose an algorithm based on *label propagation*, where the propagation follows the *independent cascade* model [Kempe et al. 2003]. A similar approach is also taken in Chen et al. [2014]: The proposed algorithm works in two phases, where the first phase discovers the community structure of the network, and the second phase uses the information of communities detect the set of possible seed influencers.

In these works, communities are only a way to reduce the search space of the problem of finding influential users, not the goal. Moreover, the setting differs considerably from ours: In their framework, both the social network and the influence strength are given in input, while in our case neither of the two is known.

Barbieri et al. [2013a] studies the following problem: Given both the social graph and the log of user activity as input, the goal is to detect communities that “explain” well the two pieces of input. In simpler terms, the idea is to do better community detection using the additional information contained in the activity log. Towards this goal, Barbieri et al. propose the Community-Cascade Network (CCN) model, a stochastic mixture membership generative model that can fit, at the same time, the social graph and the log of user activity.

Mehmood et al. [2013] introduces a model for analyzing information propagation and social influence at the granularity of communities. The analysis of the community-level influence propagation network on a real-world dataset shows that the network is almost acyclic.

Our work also collocates itself in the intersection of community detection and social contagion, but it differs from the proposals of Barbieri et al. [2013a] and Mehmood et al. [2013], as we tackle the problem of community detection *without the network*.

3. GENERAL FRAMEWORK

In this section, we introduce our general stochastic framework, which we will instantiate with different diffusion models in Sections 4 and 5.

3.1. Preliminaries

We are given a log of past user activity $\mathbb{D}$ defined as a relation $(User, Item, Time)$ where each tuple (u, i, t) represents the fact that the node u adopted the item i at the time t. We let V denote the set of all users, that is, the projection of $\mathbb{D}$ over the first column, and $\mathcal{I}$ denote the universe of items, that is, the projection of $\mathbb{D}$ on the second column, and we assume the time is an integer $t \in [0, T)$. We also use D_i to denote the overall activity on item i, that is, the selection of the tuples of $\mathbb{D}$ where $Item = i$. We call it the *propagation trace* of i. The projection of D_i on the first column is denoted as C_i. We assume $|\mathbb{D}| = L$, $|V| = M$, and $|\mathcal{I}| = N$. Furthermore, we denote $|C_i| = |D_i| = L_i$ as the size of the propagation trace of i and $N_u = |\{i|u \in C_i\}|$ as the number of items adopted by user u.

Let $t_u(i)$ represent the adoption time of the user u for the trace D_i; with $t_u(i) = \infty$ if u does not adopt i by time T. We denote the time delay between the adoption of two users u, v on the item i as $\Delta_{u,v}(i) = t_u(i) - t_v(i)$. We also define $\Delta_u(i) = T - t_u(i)$. When i is clear from the context, we simply write $\Delta_{u,v}$ and Δ_u. Finally, let $C_{i,t}$ denote the set of users active on the trace i by time t, i.e., $C_{i,t} = \{u \in V : t_u(i) < t\}$. For notational convenience, we shall also denote $C_{i,t_u(i)}$ as $C_{i,u}$, for each $u \in C_i$. Finally, we denote by $u \prec_i v$ the fact that both u and v are in C_i, and $t_u(i) \le t_v(i)$. Analogously, $u \succ_i v$ holds when both u and v are in C_i, and $t_u(i) > t_v(i)$ (hence v is a possible influencer for u's adoption). With an abuse of notation, we denote $u \preceq_i v$ when either $u = v$ or $u \prec_i v$.

3.2. Framework Overview

Given only a log $\mathbb{D}$ of users' activities, our goal is to detect communities in an unobserved network whose set of nodes correspond to the set of users V of $\mathbb{D}$. By communities we mean—as usual in the literature—clusters of nodes of a social network that exhibit high internal and low external link density. For ease of presentation, we talk about communities as a complete partitioning of V, that is, communities are disjoint. However, it is worth noting that, being stochastic, our method produces for each user the level of membership in each community: Therefore a soft assignment to multiple overlapping communities is always possible. While detecting communities, we also aim to learn for each community which are the most influential users, that is, those users who are most likely to influence the rest of the community to adopt a certain item i.

One possible approach is to forget about the existence of an underlying unobserved social network. Instead, we just tackle the problem with a standard clustering approach: V is the set of objects to be clustered, and the actions of each user in V (which item i is adopted and at which time t) is its description. One main drawback of this approach is that it provides only clusters of users and no information about the influence of users in their respective cluster.

Another possible approach is to focus on social influence to infer the social network from the user activity log $\mathbb{D}$, following the framework of Gomez Rodriguez et al. [2010, 2011]. Then we can apply some standard community detection algorithm to the resulting "reconstructed" network. As previously discussed, this approach has several drawbacks, the main one being the runtime quadratic in the size of V.

We can consider these two approaches as the two opposite extremes of the spectrum[5]: one totally ignoring the existence of the network effect and the other one explicitly reconstructing the network. Our proposal collocates between these two extremes: Although it does not attempt a direct reconstruction of the network, it assumes that information spreads over social connections. Thus the assumption behind our framework is that the unobserved network naturally shapes the process of information diffusion.

A high-level overview of our framework is as follows:

—We assume the existence of an unobserved social network having a modular structure (as typical of social networks). This means that communities exist, and they are densely connected internally and loosely connected with other communities.
—We assume that the process of the adoption of items is governed by an underlying stochastic diffusion process over the unobserved social network. In particular, the diffusion model is based on community-level influence.
—Each user is associated with a level of membership and a level of influence in each community. These are the parameters of the diffusion model that we need to learn. The adoption of an item i by a user u depends on the level of adoption of the item in the community of u or, in other words, by the influence exerted by the other members of the community on u for adopting i.
—By fitting the model parameters to the user activity log $\mathbb{D}$, we learn the community membership and influence levels.

This general framework can be instantiated by using different stochastic (community-level) diffusion models, leading to different community detection methods. In this work we study two such models in Section 4 and 5, respectively. We conclude this section by presenting the EM-like algorithm for fitting the model parameters to the user activity log.

3.3. Modeling Maximum Likelihood

We base our community detection algorithm on a probabilistic framework where we assume the existence of a latent association of each user with a given community, which governs both her/his social ties and her/his attitude towards item adoptions. Social ties are not observed in our framework. However, they are dense within the same community. Thus, by modeling the (observed) propagation behavior, we can still infer such ties as well as the underlying latent community. We resort to mixture modeling here, which has been already proven as a flexible and powerful framework for overlapping community detection [Newman and Leicht 2007; Ren et al. 2009; Davis and Carley 2008]. However, our model parameterizes the probability of actions propagating due to influence, whereas traditional approaches where the network structure is known focus on modeling the probability of edges. According to such an approach, we are still able to detect communities, and, for each community, we can also detect the contribution of each node to the propagations.

We assume that each propagation trace is independent from the others, and we adopt a maximum a posteriori perspective. That is, we hypothesize that action probabilities adhere to a mathematical model governed by a set of parameters Θ. The likelihood of the data given the model parameters Θ can hence be expressed as

$$\mathcal{L}(\Theta; \mathbb{D}) = \prod_{i \in \mathcal{I}} P(D_i|\Theta),$$

where $P(D_i|\Theta)$ represents the likelihood to observe the propagation trace behavior relative to i in $\mathbb{D}$. This can be deemed relative to the contribution of each user in it. In

[5]In our experimental assessment (Section 6), we compare our proposal against these alternative approaches.

practice, we can devise a Markov chain of probabilities relative to the specific adoptions (or non-adoptions),

$$P(D_i|\Theta) = \prod_{u\in V} P(a_{u,i}|D_{u,i}, \Theta),$$

where $a_{u,i}$ is the action $(u, i, t_u(i))$, and $D_{u,i}$ is the sequence of all actions in D_i that occur prior to $t_u(i)$. In principle, there is no independence assumption among actions, since an action occurring at time t may depend on other actions that occurred on the same item prior to t.

The corresponding learning problem is finding the optimal $\hat{\Theta}$ that maximizes $\mathcal{L}(\Theta; \mathbb{D})$. Following the standard mixture modeling approach [Dempster et al. 1977], we assume that the adoptions of each user can be explained by the community he/she belongs to. That is, we assume that a hidden binary variable $z_{u,k}$ denotes the membership of user u to community k, with the constraint $\sum_{k=1}^{K} z_{u,k} = 1$. Thus, Θ can be partitioned into $\{\pi_1, \ldots, \pi_K, \Theta_1, \ldots, \Theta_K\}$, where Θ_k represents the parameter set relative to community k, and $\pi_k = P(z_{u,k} = 1)$ is the probability that a user is associated with community k. We can rewrite the likelihood as

$$\mathcal{L}(\Theta; \mathbb{D}) = \prod_{i\in\mathcal{I}} \prod_{u} \sum_{k=1}^{K} P(a_{u,i}|D_{u,i}, \Theta_k)\pi_k,$$

which can be optimized by resorting to the traditional EM algorithm. We rewrite the complete likelihood as

$$P(\mathbb{D}, \mathbf{Z}, \Theta) = P(\mathbb{D}|\mathbf{Z}, \Theta) \cdot P(\mathbf{Z}|\Theta) \cdot P(\Theta), \tag{1}$$

where

$$\begin{aligned} P(\mathbb{D}|\mathbf{Z}, \Theta) &= \prod_{u\in V} \prod_{k=1}^{K} P(u|\Theta_k)^{z_{u,k}} \\ P(u|\Theta_k) &= \prod_{i} P(a_{u,i}|D_{u,i}, \Theta_k) \\ P(\mathbf{Z}|\Theta) &= \prod_{u\in V} \prod_{k=1}^{K} \pi_k^{z_{u,k}}, \end{aligned}$$

and $P(\Theta)$ represents the prior relative to the parameter set Θ. Inspired by Figueiredo and Jain [2002], we choose to model the latter as

$$P(\Theta) \propto \prod_{k=1}^{K} \pi_k^{-\frac{1}{2}\sqrt{|\Theta_k|}},$$

with the interpretation that, for fixed K, the parameters π_k allow an "improper" Dirichlet-type prior. This enables a formulation of EM algorithm that allows the automatic detection of the optimal number K of communities. By standard manipulation of Equation (1), the *Complete-Data Expectation Likelihood* [Dempster et al. 1977] is given by

$$\begin{aligned} \mathcal{Q}(\Theta; \Theta') =& E[\log P(\mathbb{D}, \mathbf{Z}, \Theta)|\mathbb{D}; \Theta'] \\ &\propto \sum_{u\in V} \sum_{k=1}^{K} \gamma_{u,k}\{\log P(u|\Theta_k) + \log \pi_k\} - \sum_{k=1}^{K} \frac{N_k}{2} \log \pi_k, \end{aligned} \tag{2}$$

where $N_k = \sqrt{|\Theta_k|}$ and $\gamma_{u,k} \equiv P(z_{u,k} = 1|u, \Theta')$. Optimizing $\mathcal{Q}(\Theta; \Theta')$ with respect to π_k under the constraints $\sum_k \pi_k = 1, 0 \leq \pi_k \leq 1$ yields:

$$\pi_k = \frac{\max\left\{0, \sum_{u \in V} \gamma_{u,k} - N_k/2\right\}}{\sum_{k=1}^{K} \max\left\{0, \sum_{u \in V} \gamma_{u,k} - N_k/2\right\}}. \tag{3}$$

Here, the proposed prior allows an adjustment to the estimation of the π_k parameters which enables "annihilation": A community not supported by a sufficient number of users is removed. Thus, we can start with an arbitrarily large initial number of communities and then infer the final number K by letting some of the mixing probabilities π_k be zero.

ALGORITHM 1: Learning Algorithm

```
   Input: Propagation log $\mathbb{D}$, and $K_{max} \in \mathbb{N}^+$.
   Output: Optimal $K$ value; the set of all parameters, $\Theta$.
1  Arbitrarily initialize $\Theta^{(0)}$ // Initialization of model parameters
2  $K \leftarrow K_{max}$;
3  repeat
4      forall the $k$ such that $\pi_k > 0$ do
5   E-step  forall the $u$ do
6               compute $\gamma_{u,k} = \frac{P(u|\Theta_k^{(t-1)})\pi_k}{\sum_{k=1}^{k} P(u|\Theta_k^{(t-1)})\pi_k}$
7           end
8      end
9      forall the $k$ such that $\pi_k > 0$ do
10  M-step  compute $\pi_k = \frac{\max\{0, \sum_u \gamma_{u,k} - N_k/2\}}{\sum_{h=1}^{k} \max\{0, \sum_u \gamma_{u,h} - N_h/2\}}$
11          Normalize $\{\pi_1, \ldots, \pi_K\}$;
12          if $\pi_k > 0$ then
13              compute additional model components;
14              compute $\Theta_k^{(t)} = \arg\max_k \mathcal{Q}(\Theta; \Theta^{(t-1)})$;
15          end
16          else
17              $K \leftarrow K - 1$;
18          end
19     end
20
21 until Convergence;
```

The general EM scheme is shown in Algorithm 1. As discussed in Figueiredo and Jain [2002], a further advantage of the scheme is its robustness to random initialization: By starting with an arbitrarily large number of components, we can avoid the pitfalls of local maxima, since the whole parameter space is likely to be covered. As a side note, the modeling of the prior $P(\Theta)$ is a major difference w.r.t. Figueiredo and Jain [2002]: When $|\Theta_k|$ is of the same order of magnitude as $|V|$, the original formulation of the prior in Figueiredo and Jain [2002] would produce an underestimation of the number of communities. Furthermore, reducing the weight of $|\Theta_k|$ in the computation of π_k allows us to reformulate the algorithm without optimizing the components in sequence, which would require a prohibitive computational cost for large $\mathbb{D}$.

The above modeling is a general framework that is parametric to the component $P(u|\Theta_k)$. In turn, the latter depends on the way we model the probability

$P(a_{u,i}|D_{u,i}, \Theta_k)$ for each adoption $a_{u,i} \equiv (u, i, t)$. We explore two different ways of modeling $P(a_{u,i}|D_{u,i}, \Theta_k)$, which focus on two different perspectives.

(1) The probability that u adopts i is the result of a Bernoullian process on i, that is, $P(a_{u,i}|D_{u,i}, \Theta_k) \equiv P(i|u, t, D_{u,i}, \Theta_k)$, and time proceeds in discrete steps.
(2) The final model does not consider *whether* a user adopts i but *when* the adoption happens, that is, $P(a_{u,i}|D_{u,i}, \Theta_k) \equiv P(t|i, u, D_{u,i}, \Theta_k)$.

We next explore each strategy in turn. We consider a binary and progressive activation process: At a given timestamp, each user is either active or inactive, and active users cannot become inactive again.

4. COMMUNITY-LEVEL INDEPENDENT CASCADE MODEL

In the first alternative, we assume a Bernoullian model for users' adoptions of items. As a result, the likelihood $P(u|\Theta_k)$ can be specified over the observed binary data $Y_{i,u}$, where $Y_{i,u} = 1$ if $u \in C_i$, and $Y_{i,u} = 0$ otherwise.

When the social relationships are explicit, it is possible to define a propagation model that describes how adoptions spread across the network [Kempe et al. 2003] and to model information propagation and community structure suitably [Barbieri et al. 2013a]. In these models, a user's tendency to become active increases monotonically as more of its social peers become active. We next adapt this concept to a network-oblivious situation, where we assume that the user's tendency to become active depends on the influence exerted within the community of membership.

The *Community-Independent Cascade (C-IC)* model draws from the Independent Cascade model (IC) [Kempe et al. 2003] and models the idea that each user exerts the same degree influence over members of each community. Time unfolds in discrete timestamps. As in IC, when a user v becomes active, say, at time t, it is considered contagious and has a single chance of influencing each inactive neighbor u, independently of the history thus far. The IC model specifies pairwise influence probabilities $p_{v,u}$, which express the likelihood of success for v's attempt in activating her/his neighbor u. Here, since the network information is not available, we assume that v exerts her/his influence "globally," with a strength $p_v^k \in [0, 1]$ which depends on the community k of the targeted node. The idea is that the community-level influence of each user v is higher in the community she/he belongs to. According to this principle, we assume that information mainly propagate locally and spread across communities thanks to the presence of users who exhibit high degree of "external" influence.

Following Mathioudakis et al. [2011], we adopt a delay threshold Δ to define influencers. Specifically, we define $\mathcal{F}_{i,u}^{+}$ as the set of users who potentially influenced u in the adoption of i:

$$\mathcal{F}_{i,u}^{+} = \{v \in V | 0 \leq t_u(i) - t_v(i) \leq \Delta\}.$$

The set $\mathcal{F}_{i,u}^{-}$ of users who definitely failed in influencing u over i is defined similarly:

$$\mathcal{F}_{i,u}^{-} = \{v \in V | t_u(i) - t_v(i) > \Delta\}.$$

Then, we can specify $P(u|\Theta_k)$ as

$$P(u|\Theta_k) = \prod_i P_{+}(i|u, \Theta_k)^{Y_{i,u}} \cdot P_{-}(i|u, \Theta_k), \tag{4}$$

where $P_{+}(i|u, \Theta_k)$ represents the probability that some of the potential influencers activated u and $P_{-}(i|u, \Theta_k)$ the probability that none of the "out-of-react" influencers succeeded:

$$P_+(i|u,\Theta_k) = 1 - \prod_{v\in\mathcal{F}^+_{i,u}} \left(1-p^k_v\right) \qquad P_-(i|u,\Theta_k) = \prod_{v\in\mathcal{F}^-_{i,u}} \left(1-p^k_v\right).$$

When $Y_{i,u} = 0$, the set $\mathcal{F}^+_{i,u}$ is empty, thus neutralizing $P_+(i|u,\Theta_k)$. As a consequence, we can omit the exponent in the first component and specify the complete-data likelihood as

$$P(\mathbb{D}|\mathbf{Z},\Theta) = \prod_{i,u,k} \left[1 - \prod_{v\in\mathcal{F}^+_{i,u}} \left(1-p^k_v\right)\right]^{z_{u,k}} \cdot \left[\prod_{v\in\mathcal{F}^-_{i,u}} \left(1-p^k_v\right)\right]^{z_{u,k}}.$$

4.1. Learning Influence Weights

The analytical optimization of $\mathcal{Q}(\Theta;\Theta^{(t-1)})$ is still difficult. We resort to the explicit modeling of the influencers as hidden data to simplify the optimization procedure. That is, let $w_{i,u,v}$ be a binary variable such that $w_{i,u,v} = 1$ if v triggered the adoption of the item i by u, and let $\mathbf{W}$ denote the set of all possible $w_{i,u,v}$ such that $v \in \mathcal{F}^+_{i,u}$. Then, we can rewrite the complete-data likelihood relative to $\mathbf{W}$ as

$$P(\mathbb{D},\mathbf{Z},\mathbf{W},\Theta) = P(\mathbb{D},\mathbf{W}|\Theta,\mathbf{Z}) \cdot P(\mathbf{Z}|\Theta) \cdot P(\Theta),$$

where

$$P(\mathbb{D},\mathbf{W}|\Theta,\mathbf{Z}) = \prod_{i,u,k}\prod_{v\in\mathcal{F}^-_{i,u}} \left(1-p^k_v\right)^{z_{u,k}} \prod_{i,u,k}\prod_{v\in\mathcal{F}^+_{i,u}} \left(p^k_v\right)^{w_{i,u,v}\cdot z_{u,k}} \left(1-p^k_v\right)^{(1-w_{i,u,v})\cdot z_{u,k}}.$$

As a consequence, the contribution to $\mathcal{Q}(\Theta;\Theta^{(t-1)})$ in the second row of Equation (2) can be rewritten as

$$\sum_u\sum_k \gamma_{u,k}\left(\log\pi_k + \sum_i\sum_{v\in\mathcal{F}^-_{i,u}} \log\left(1-p^k_v\right) + \sum_i\sum_{v\in\mathcal{F}^+_{i,u}} \eta_{i,u,v,k}\log p^k_v + (1-\eta_{i,u,v,k})\log\left(1-p^k_k\right)\right),$$

where $\eta_{i,u,v,k}$ is the "responsibility" of the user v in triggering u's adoption in the context of the community k:

$$\eta_{i,u,v,k} = P\left(w_{i,u,v} = 1|u,i,z_{u,k}=1,\Theta^{(t-1)}\right) = \frac{p^k_v}{1-\prod_{w\in\mathcal{F}^+_{i,u}}\left(1-p^k_w\right)}.$$

Finally, optimizing $\mathcal{Q}(\Theta;\Theta^{(t-1)})$ with respect to p^k_v yields

$$p^k_v = \frac{\sum_{\substack{\langle u,i\rangle \\ v\in\mathcal{F}^+_{i,u}}} \gamma_{u,k}\cdot\eta_{i,u,v,k}}{S^+_{v,k}+S^-_{v,k}}, \tag{5}$$

with $S^+_{v,k} = \sum_{\substack{\langle u,i\rangle \\ v\in\mathcal{F}^+_{i,u}}} \gamma_{u,k}$ and $S^-_{v,k} = \sum_{\substack{\langle u,i\rangle \\ v\in\mathcal{F}^-_{i,u}}} \gamma_{u,k}$.

4.2. Complexity Analysis

The general scheme of the EM algorithm iterates through two steps, which (i) compute the posterior likelihood of the model parameters, given the data and (ii) update the parameters, given the posterior likelihood. The complexity of the algorithm strongly relies on efficiently implementing such steps. The posterior relies on being able to efficiently computing the likelihood of a trace. Also, in our model, we parametrize the probability of actions propagating due to influence: That is, the model parameters represent the influence of each user within each community, and item adoption is modeled as the effect of such an influence.

An apparent difficulty in this modeling strategy stems in the latent nature of an influencer in an adoption: Since it is not known, both the likelihood of a trace and the parameters exhibit a stochastic dependency towards the set of all possible eligible influencers. Fortunately, the adopted modeling allows us to devise a strategy for cumulating the contribution of each possible influencer independently of the specific item adoption. This guarantees that both the likelihoods of traces and the updates of the model parameters can be accomplished in just two scans of the log trace.

To see why, we can analyze the complexity of the learning phase for C-IC by considering the E and M steps of the EM algorithm separately (see Algorithm 1). For the E step, we need to compute $\Pr(u|\Theta_k)$ for each user u and community k. We need to consider two cases concerning Δ, which is the delay threshold used to define potential influencers.

LEMMA 4.1. *If $\Delta = \infty$, then*

$$\log \Pr(u|\Theta_k) = A_{u,k} + B_k - B_{u,k},$$

where $A_{u,k}$, B_k, and $B_{u,k}$ are as defined in Table I .

PROOF. We can rewrite the log probability of observing the adoptions of a user within the community k as

$$\begin{aligned}\log \Pr(u|\Theta_k) &= \sum_{i:u\in C_i} \log\left\{1 - \prod_{v\prec_i u}\left(1-p_v^k\right)\right\} + \sum_{i:u\notin C_i}\sum_{v\in C_i}\log\left(1-p_v^k\right)\\ &= \sum_{i:u\in C_i}\log(1-A_{i,u,k}) + \sum_{i:u\notin C_i} B_{i,k}\\ &= A_{u,k} + B_k - B_{u,k}. \quad \square\end{aligned}$$

COROLLARY 4.2. *When $\Delta = \infty$, the complexity of the E step is $O(KL + KM)$ in time and $O(KM + KN)$ in space.*

PROOF. Overall, the computation of all $\log \Pr(u|\Theta_k)$ requires two scans over $\mathbb{D}$ and K; then, the updated values of $\gamma_{u,k}$ can be computed by transforming those logs into probabilities. In the first scan, the components $A_{i,u,k}$, $A_{u,k}$, and $B_{i,k}$ can be computed incrementally, without the need of storing the values for $A_{i,u,k}$. Assuming that v is the user directly preceding u in the trace i, it holds that $A_{i,u,k} = A_{i,v,k} \cdot (1 - p_v^k)$. $A_{u,k}$ can be computed by summing up the components $A_{i,u,k}$ for each trace i that involves the user u. The same holds for $B_{i,k}$, which is computed when iterating over trace i and the community k. Finally, after computing $B_{i,k}$, we can obtain the components $B_{u,k}$ for all users u with a second scan over $\mathbb{D}$ and K. $\square$

For the case $\Delta < \infty$, the main problem is efficiently computing $\mathcal{F}^+_{i,u}$ and $\mathcal{F}^-_{i,u}$ for each user u and item i. Let $l_{i,v}$ be the user $u \in \mathcal{F}^-_{i,v}$ with the shortest temporal gap $t_v(i) - t_u(i)$

Table I. Definition of the Counters Used for the C-IC Model (Section 4.2)

counter	*definition*	*description*
N_v	$\lvert\{i \in \mathcal{I} \mid v \in C_i\}\rvert$	Total number of items adopted by v
$A_{u,k}$	$\sum_{i:u\in C_i} \log(1 - A_{i,u,k})$	Log-likelihood of u's activations when the community is k
$A_{i,u,k}$	$\begin{cases} \prod_{v \prec_i u} (1 - p_v^k) & \Delta = \infty \\ \frac{B_{i,u,k}}{B_{i,l_{i,u},k}} & \Delta < \infty \end{cases}$	Likelihood that all influencers fails to trigger u's activation in the trace i under community k
B_k	$\sum_i B_{i,k}$	Cumulative over all traces i of $B_{i,k}$
$B_{u,k}$	$\sum_{i:u\in C_i} B_{i,k}$	Cumulative over all trace i on which u is active of $B_{i,k}$
$B_{i,k}$	$\sum_{v\in C_i} \log\left(1 - p_v^k\right)$	Log-likelihood that all active nodes in trace i will fail to trigger a next activation under community k
$\tilde{B}_{u,k}$	$\sum_{i:u\in C_i} \log B_{i,l_{i,u},k}$	Cumulative over all trace i on which u is active of $B_{i,w,k}$ where $w = l_{i,u}$
$B_{i,u,k}$	$\prod_{v \preceq_i u} (1 - p_v^k)$	Likelihood that all the influencers up to u will fail to trigger the next activation in trace i under community k
$C_{v,k}$	$\begin{cases} \sum_{i:v\in C_i} C_{i,k} & \Delta = \infty \\ \sum_{i:v\in C_i} C_{i,v,k} & \Delta < \infty \end{cases}$	Cumulative over all traces i on which v is active of $C_{i,k}$ (resp. $C_{i,v,k}$)
$D_{v,k}$	$\begin{cases} \sum_{i:v\in C_i} C_{i,v,k} & \Delta = \infty \\ \sum_{i:v\in C_i} C_{i,s_{i,v},k} & \Delta < \infty \end{cases}$	Cumulative over all traces i on which v is active of $C_{i,v,k}$ (resp. $C_{i,s_{i,v},k}$)
$C_{i,v,k}$	$\begin{cases} \sum_{u \preceq_i v} \frac{\gamma_{u,k}}{1-A_{i,u,k}} & \Delta = \infty \\ \sum_{u \prec_i v} \frac{\gamma_{u,k}}{1-A_{i,u,k}} & \Delta < \infty \end{cases}$	Cumulative over influencers for v's activation in trace i of their conditional community membership given that their activations happen in community k
$C_{i,k}$	$\sum_{u\in C_i} \frac{\gamma_{u,k}}{1 - A_{i,u,k}}$	Cumulative over active users in the trace i of conditional community memberships $\gamma_{u,k}$ given that their activations happen in community k
$\Gamma_{i,v,k}$	$\sum_{u \preceq_i v} \gamma_{u,k}$	Cumulative involvement in the community k of potential influencers for user v (included) on trace i
Γ_k	$\sum_u \gamma_{u,k}$	Cumulative over u of $\gamma_{u,k}$: this acts as a prior for community k.
$\Gamma_{v,k}$	$\sum_{i:v\in C_i} \Gamma_{i,v,k}$	Cumulative over all users v active on trace i of $\Gamma_{i,v,k}$
$\Gamma_{i,k}$	$\sum_{u\in C_i} \gamma_{u,k}$	Cumulative over users u active in the trace i of $\gamma_{u,k}$.

from v in the adoption of i. Dually, let $s_{i,v}$ be the user u such that $v \in F^{+}_{i,u}$ with the farthest distance $t_u(i) - t_v(i)$. Notice that $s_{i,v}$ and $l_{i,v}$ can be computed in $O(\log L_i)$ by exploiting appropriate index structures, such as, for example, B-Trees.

LEMMA 4.3. *When $\Delta < \infty$, then*

$$\log \Pr(u|\Theta_k) = A_{u,k} + B_k - B_{u,k} + \tilde{B}_{u,k},$$

where $A_{u,k}$, B_k, $B_{u,k}$, and $\tilde{B}_{u,k}$ are defined as in Table I.

Proof. We can observe the following:

$$\begin{aligned}
\log \Pr(u|\Theta_k) &= \sum_{i:u\in C_i} \log\left\{1 - \prod_{v\in \mathcal{F}^{+}_{i,u}} \left(1 - p^k_v\right)\right\} + \sum_i \sum_{v\in F^{-}_{i,u}} \log\left(1 - p^k_v\right)\\
&= \sum_{i:u\in C_i} \log(1 - A_{i,u,k}) + \sum_i \log B_{i,l_{i,u},k}\\
&= A_{u,k} + \sum_{i:u\notin C_i} B_{i,k} + \sum_{i:u\in C_i} \log B_{i,l_{i,u},k}\\
&= A_{u,k} + B_k - \sum_{i:u\in C_i} B_{i,k} + \sum_{i:u\in C_i} \log B_{i,l_{i,u},k}\\
&= A_{u,k} + B_k - B_{u,k} + \tilde{B}_{u,k}. \quad \square
\end{aligned}$$

Corollary 4.4. *When $\Delta < \infty$, the complexity of the E step is $O(L(K + \log(\max_i L_i)) + KM)$ time and $O(K(\max_i L_i) + KM + KN)$ space.*

Proof. For each element u in each trace i, we need to compute $l_{i,u}$, which takes $O(\log L_i)$. The values $B_{i,u,k}$ can be computed incrementally by exploiting the recursive relationship for each adjacent pair in a trace. Finally, we only need pre-allocate the space for the $A_{i,u,k}$ values relative to a single trace. In fact, if $v = l_{i,u}$, then we can associate with v a dual variable $\tilde{l}_{i,v}$ such that $u = \tilde{l}_{i,v}$, and hence we can assume a look-ahead strategy where the current value $B_{i,v,k}$ can be pre-stored to contribute to the computation of $A_{i,\tilde{l}_{i,v},k}$. Finally, the B_k, $B_{i,k}$, $B_{u,k}$, and $\tilde{B}_{u,k}$ components can be computed incrementally and hence $\log \Pr(u|\Theta_k)$ can be computed in at most two scans of $\mathbb{D}$. $\square$

Although not strictly required for the step E, it is convenient to store the $A_{i,u,k}$ values relative to each $(u, i) \in \mathbb{D}$. The term $\eta_{i,u,v,k}$ can be rewritten as $p^k_v/(1 - A_{i,u,k})$. Storing $A_{i,u,k}$ allows us to efficiently compute the update for the parameters p^k_v, as stated by the following lemmas.

Lemma 4.5. *Given a user u and a community k, let $\tilde{p}^k_v$ be the value of p^k_v computed in the previous iteration and assume $\Delta = \infty$. Then,*

$$p^k_v = \tilde{p}^k_v \frac{C_{v,k} - D_{v,k}}{N_v \Gamma_k - \Gamma_{v,k}},$$

where $C_{v,k}$, $D_{v,k}$, N_v, Γ_k, and $\Gamma_{v,k}$ are defined as in Table I.

Proof. We can write Equation (5) as $p^k_v = num_{v,k}/den_{v,k}$. Let us consider each element in turn. Concerning the numerator, we have

$$\begin{aligned}
num_{v,k} &= \sum_{i:v\in C_i} \sum_{u\succ_i v} \gamma_{u,k} \cdot \eta_{i,u,v,k}\\
&= \tilde{p}^k_v \sum_{i:v\in C_i} \sum_{u\succ_i v} \frac{\gamma_{u,k}}{1 - A_{i,u,k}}\\
&= \tilde{p}^k_v \sum_{i:v\in C_i} (C_{i,k} - C_{i,v,k})\\
&= \tilde{p}^k_v (C_{v,k} - D_{v,k}).
\end{aligned}$$

As for the denominator, we have

$$\begin{aligned}
den_{v,k} &= S^+_{v,k} + S^-_{v,k} \\
&= \sum_{i:v\in C_i}\sum_{u\succ_i v}\gamma_{u,k} + \sum_{i:v\in C_i}\sum_{u\notin C_i}\gamma_{u,k} \\
&= \sum_{i:v\in C_i}\left(\sum_{u\in C_i}\gamma_{u,k} - \sum_{u\preceq_i v}\gamma_{u,k}\right) + \sum_{i:v\in C_i}\left(\sum_{u}\gamma_{u,k} - \sum_{u\in C_i}\gamma_{u,k}\right) \\
&= \sum_{i:v\in C_i}(\Gamma_{i,k} - \Gamma_{i,v,k}) + \sum_{i:v\in C_i}(\Gamma_k - \Gamma_{i,k}) \\
&= N_v\Gamma_k - \Gamma_{v,k}. \quad \square
\end{aligned}$$

COROLLARY 4.6. *When $\Delta = \infty$, the complexity of the M step is $O(KL + KM)$ in time and $O(KL + KM + KN)$ in space.* $\square$

Again, a similar formulation can be devised for the case $\Delta < \infty$, which relies on the computation of $s_{i,u}$ for each trace i and user u within i.

LEMMA 4.7. *Given a user u and a community k, let $\tilde{p}^k_v$ be the value of p^k_v computed in the previous iteration and assume $\Delta < \infty$. Then,*

$$p^k_v = \tilde{p}^k_v \frac{D_{v,k} - C_{v,k}}{N_v\Gamma_k},$$

where $D_{v,k}$, $C_{v,k}$, N_v, and Γ_k are defined as in Table I.

PROOF. Again, we can analyse separately the numerator and denominator. For the numerator, we can observe the following:

$$\begin{aligned}
num_{v,k} &= \sum_{i:v\in C_i}\sum_{u:v\in F^+_{i,u}}\gamma_{u,k}\cdot\eta_{i,u,v,k} \\
&= \tilde{p}^k_v \sum_{i:v\in C_i}\sum_{u:v\in F^+_{i,u}}\frac{\gamma_{u,k}}{1 - A_{i,u,k}} \\
&= \tilde{p}^k_v \sum_{i:v\in C_i}(C_{i,s_{i,v},k} - C_{i,v,k}) \\
&= \tilde{p}^k_v(D_{v,k} - C_{v,k}).
\end{aligned}$$

Concerning the denominator, we can observe that, when $\Delta < \infty$, we have

$$\begin{aligned}
S^-_{v,k} &= \sum_{\substack{\langle u,i\rangle \\ v\in F^-_{i,u}}}\gamma_{u,k} \\
&= \sum_{i:v\in C_i}\sum_{u\notin C_i}\gamma_{u,k} + \sum_{i:v\in C_i}\sum_{u\in C_i}(1 - \mathbb{1}[\![t_u(i) \le t_v(i) + \Delta]\!])\gamma_{u,k} \\
&= \sum_{i:v\in C_i}\left(\sum_{u}\gamma_{u,k} - \sum_{u\in C_i}\gamma_{u,k}\right) + \sum_{i:v\in C_i}\sum_{u\in C_i}\gamma_{u,k} - \sum_{i:v\in C_i}\sum_{u:v\in \mathcal{F}^+_{i,u}}\gamma_{u,k} \\
&= N_v\Gamma_k - S^+_{v,k}.
\end{aligned}$$

It turns out that $den_{v,k}$ can be simplified as $N_v\Gamma_k$. $\square$

COROLLARY 4.8. *When $\Delta < \infty$, the complexity of the M step is $O(L(K + \max_i(\log L_i)) + KM)$ in time and $O(KL + KM + KN)$ in space.*

PROOF. $D_{v,k}$ can be computed in an incremental fashion. We can devise a look-ahead strategy for computing $D_{v,k}$. More specifically, we can define $\tilde{s}_{i,u}$ such that whenever $u = s_{i,v}$, then $v = \tilde{s}_{i,u}$. As a consequence, each $C_{i,v,k}$ contributes to the corresponding $D_{\tilde{s}_{i,u},k}$. A single scan of the $\mathbb{D}$ allows us to compute numerator and denominator for each p_v^k, which can be updated afterwards. □

We are finally able to state the main complexity result for the C-IC.

THEOREM 4.9. *The complexity of a single step of the EM algorithm plugged with the C-IC model is $O(KL + KM)$ in time and $O(KL + KM + KN)$ space when $\Delta = \infty$ and $O(L(K + \max_i(\log L_i) + KM)$ in time and $O(KL + KM + KN)$ in space when $\Delta < \infty$.*

The resulting linear complexity of the iteration of the EM algorithm guarantees that the approach scales to very large networks, at the same time guaranteeing the discovery of high-quality communities.

5. MODELING TEMPORAL DYNAMICS

C-IC does not explicitly model temporal dynamics, as it focuses on modeling just binary activations by employing a discrete-time propagation model. Here we present a more fine-grained modeling that exploits time to better characterize the overall diffusion process.

Given an observation window $[0, T]$, the idea is to explicitly model the likelihood of the time at which each user adopted each item or the likelihood that the considered adoption did not happen within time T. This approach assumes that there is a dependency between the adoption time of the influencer and the one of the influenced. In NetRate [Gomez Rodriguez et al. 2011], previously described in Section 2, this dependency in modeled by a conditional likelihood $f(t_u|t_v, \alpha_{v,u})$ of transmission, which depends on the delay $\Delta_{v,u}$. The likelihood of a propagation can be formulated by applying standard survival analysis [Lee and Wang 2003], in terms of survival $S(t_u|t_v, \alpha_{v,u})$ (modeling the probability that a user survives uninfected at least until time t_u) and hazard functions $H(t_u|t_v, \alpha_{v,u})$ (modeling instantaneous infections).

We reformulate this framework into a community-based scenario. The *Community-Rate* (C-Rate) propagation model is characterized by the following assumptions:

- User's influence is limited to the community she/he belongs to. That is, the user is likely to influence/be influenced by members of the same community, while the effect of influence is marginal on members of a different community.
- Each user exhibits the same degree transmission rate on members of the same community k. That is, the information diffusion from the user v to u within the k-th community is characterized by the density $f(t_u|t_v, \alpha_{v,k})$, where $\alpha_{v,k}$ is related to the expected delay on the activations that v triggers within community k. The probability of contagion depends on the time delay $\Delta_{v.u}$.

The parameter $\alpha_{v,k}$ has a direct interpretation in terms of influence: High values of $\alpha_{v,k}$ cause short delays, and, as a consequence, they denote v as strongly influential within k.

On the basis of the above observations, we can adapt the NetRate model to fit the scheme of Section 3 by plugging

$$P(u|\Theta_k) = \prod_{i:u\notin C_i} \prod_{v\in C_i} S(T|t_v(i), \alpha_{v,k}) \cdot \prod_{i:u\in C_i} \prod_{v\prec_i u} S(t_u(i)|t_v(i), \alpha_{v,k}) \sum_{v\prec_i u} H(t_u(i)|t_v(i), \alpha_{v,k}). \tag{6}$$

5.1. Learning

Again, instead of directly optimizing the likelihood based on Equation (6) above, we introduce the latent binary variable $w_{i,u,v}$, denoting the fact that u has been infected by v on i. Then, the likelihood can be rewritten by defining

$$P(\mathbb{D}, \mathbf{W}|\mathbf{Z}, \Theta) = \prod_{\langle u,i\rangle\notin\mathbb{D}} \prod_k \prod_{v\in C_i} S(T|t_v(i), \alpha_{v,k})^{z_{u,k}} \cdot \prod_{\langle u,i\rangle\in\mathbb{D}} \prod_k \prod_{v\prec_i u} H(t_u(i)|t_v(i), \alpha_{v,k})^{w_{i,u,v}z_{u,k}} \cdot S(t_u(i)|t_v(i), \alpha_{v,k})^{z_{u,k}}$$

and replacing $P(\mathbb{D}|\mathbf{Z}, \Theta)$ with the above component in the likelihood. In the following, we adopt the exponential distribution $f(t_u|t_v, \alpha_{v,k}) = \alpha_{v,k}\exp\{-\alpha_{v,k}\Delta_{v,u}\}$, for which survival and hazard can be expressed as $S(t_u|t_v, \alpha_{v,k}) = \exp\{-\alpha_{v,k}\Delta_{v,u}\}$ and $H(t_u|t_v, \alpha_{v,k}) = \alpha_{v,k}$.[6] Then, the contribution to $\mathcal{Q}(\Theta; \Theta^{(t-1)})$ in Equation (2) becomes

$$\sum_{u,k} \gamma_{u,k}\log\pi_k - \sum_{\langle u,i\rangle\notin\mathbb{D}} \sum_k \sum_{v\in C_i} \gamma_{u,k}\Delta_v\alpha_{v,k} + \sum_{\langle u,i\rangle\in\mathbb{D}} \sum_k \sum_{v\prec_i u} \eta_{i,u,v,k}\gamma_{u,k}\log\alpha_{v,k} - \sum_{\langle u,i\rangle\in\mathbb{D}} \sum_k \sum_{v\prec_i u} \gamma_{u,k}\Delta_{u,v}\alpha_{v,k},$$

and the probability of observing v as an influencer on u, i is given by

$$\eta_{i,u,v,k} = \frac{H(t_u(i)|t_v(i), \alpha_{v,k})}{\sum_{v'\prec_i u} H(t_u(i)|t_{v'}(i), \alpha_{v',k})} = \frac{\alpha_{v,k}}{\sum_{v'\prec_i u}\alpha_{v',k}}.$$

Finally, optimizing $\mathcal{Q}(\Theta; \Theta^{(t-1)})$ yields

$$\alpha_{v,k} = \frac{\sum_{\substack{\langle u,i\rangle\in\mathbb{D}\\ v\prec_i u}} \eta_{i,u,v,k}\gamma_{u,k}}{\sum_{\substack{\langle u,i\rangle\notin\mathbb{D}\\ v\in C_i}} \gamma_{u,k}\Delta_v + \sum_{\substack{\langle u,i\rangle\in\mathbb{D}\\ v\prec_i u}} \gamma_{u,k}\Delta_{u,v}}, \tag{7}$$

which expresses that the expected delay induced by v on adoptions of members of the community k depends: (i) on the ability of the user in triggering adoptions within k and (ii) on the likelihood of these adoptions to happen in the context of the considered community.

5.2. Complexity Analysis

There are many similarities between the E and M steps of C-Rate and those of C-IC with $\Delta = \infty$. Not surprisingly, the same complexity results hold, as we shall see in the following.

LEMMA 5.1. *The following relationship holds (the counters are defined in Table II).*

$$\log P(u|\Theta_k) = B_k - B_{u,k} - T(A_k - A_{u,k}) + \tilde{B}_{u,k} - \tilde{A}_{u,k} + A^l_{u,k}.$$

[6]Similar formulations can be obtained by adopting different probability functions.

Table II. Definition of the Counters Used for the C-Rate Model (Section 5.2)

counter	*definition*	*description*
A_k	$\sum_{i} A_{i,k}$	Cumulative over all traces i of $A_{i,k}$
$A_{u,k}$	$\sum_{i:u\in C_i} A_{i,k}$	Cumulative over all traces i on which u is active of $A_{i,k}$
$A_{i,k}$	$\sum_{v\in C_i} \alpha_{v,k}$	Cumulative over all active users v active on trace i of their hazard in community k
$A_{i,u,k}$	$\sum_{v\prec_i u} \alpha_{v,k}$	Cumulative over potential influencers for the activatation of u in the trace i (u included) of hazards in the community k
$\tilde{A}_{u,k}$	$\sum_{i:u\in C_i} t_u(i)A_{i,u,k}$	Cumulative over users u active in the trace i of the product between $A_{i,u,k}$ and the respective activation time
$A^l_{u,k}$	$\sum_{i:u\in C_i} \log A_{i,u,k}$	Cumulative over users u active in the trace i of the log of $A_{i,u,k}$
B_k	$\sum_{i} B_{i,k}$	Cumulative over all the traces i of $B_{i,k}$
$B_{u,k}$	$\sum_{i:u\in C_i} B_{i,k}$	Cumulative over traces i on which u is active of $B_{i,k}$
$B_{i,k}$	$\sum_{v\in C_i} \alpha_{v,k}t_v(i)$	Cumulative over users v active on trace i of the product between corresponding hazard in the community k and their activation time
$B_{i,u,k}$	$\sum_{v\prec_i u} \alpha_{v,k}t_v(i)$	Cumulative over potential influencers for the activation of u in the trace i in the community k of the product between hazards and activation times
$\tilde{B}_{u,k}$	$\sum_{i:u\in C_i} B_{i,u,k}$	Cumulative over users u active on trace i of $B_{i,u,k}$
$C_{v,k}$	$\sum_{i:v\in C_i} C_{i,k}$	Cumulative over traces i on which v is active of $C_{i,k}$
$D_{v,k}$	$\sum_{i:v\in C_i} D_{i,v,k}$	Cumulative over traces i on which v is active of $D_{i,v,k}$
$C_{i,k}$	$\sum_{u\in C_i} \gamma_{u,k}/A_{i,u,k}$	Cumulative over active users in the trace i of the ratio between conditional community memberships $\gamma_{u,k}$ and $A_{i,u,k}$
$D_{i,v,k}$	$\sum_{u\preceq_i v} \gamma_{u,k}/A_{i,u,k}$	Cumulative over influencers u for v's activation in trace i (v included) of the ratio between their community membership and $A_{i,u,k}$
Γ_k	$\sum_{u} \gamma_{u,k}$	Cumulative over u of $\gamma_{u,k}$: this acts as a prior for community k
$\Gamma_{v,k}$	$\sum_{i:v\in C_i} \Gamma_{i,k}$	Cumulative over all users v active on trace i of $\Gamma_{i,v,k}$
$\Gamma_{i,k}$	$\sum_{u\in C_i} \gamma_{u,k}$	Cumulative over users u active in the trace i of $\gamma_{u,k}$
$\Psi_{v,k}$	$\sum_{i:v\in C_i} t_v(i)\Gamma_{i,v,k}$	Cumulative over all trace i on which v is active of the product between their activation time and $\Gamma_{i,v,k}$
τ_v	$\sum_{i:v\in C_i} t_v(i)$	Cumulative over active users v on trace i of their time of adoption
$\Gamma_{i,v,k}$	$\sum_{u\preceq_i v} \gamma_{u,k}$	Cumulative involvment in the community k of potential influencers for v' activation (included) on trace i
$E_{i,k}$	$\sum_{u\in C_i} \gamma_{u,k}t_u(i)$	Cumulative over active users u on trace i of the product of community-level membership and activation time
$E_{v,k}$	$\sum_{i:v\in C_i} E_{i,k}$	Cumulative over traces i on which v is active of $E_{i,k}$
$F_{v,k}$	$\sum_{i:v\in C_i}\sum_{u\preceq_i v} \gamma_{u,k}t_u(i)$	Cumulate over potential influencers u for each active user v (included) in trace i of the product between their community membership and activation time

Proof. We can observe the following:

$$\begin{aligned}
\log P(u|\Theta_k) &= \sum_{i:u\notin C_i}\sum_{v\in C_i}\alpha_{v,k}t_v(i) - T\sum_{i:u\notin C_i}\sum_{v\in C_i}\alpha_{v,k}\\
&\quad+\sum_{i:u\in C_i}\sum_{v\prec_i u}\alpha_{v,k}t_v(i) - \sum_{i:u\in C_i} t_u(i)\sum_{v\prec_i u}\alpha_{v,k}\\
&\quad+\sum_{i:u\in C_i}\log\sum_{v\prec_i u}\alpha_{v,k}\\
&= B_k - B_{u,k} - T(A_k - A_{u,k})\\
&\quad+\sum_{i:u\in C_i} B_{i,u,k} - \sum_{i:u\in C_i} t_u(i)A_{i,u,k} + \sum_{i:u\in C_i}\log A_{i,u,k}\\
&= B_k - B_{u,k} - T(A_k - A_{u,k}) + \tilde{B}_{u,k} - \tilde{A}_{u,k} + A^l_{u,k}. \quad \square
\end{aligned}$$

Corollary 5.2. *The complexity of the E step is $O(KL+KM)$ in time and $O(KM+KN)$ in space.*

Proof. It is easy to verify that, for each trace i and for each adjacent pair u, v, the relationships $A_{i,v,k} = A_{i,u,k} + \alpha_{v,k}$ and $B_{i,v,k} = B_{i,u,k} + \alpha_{v,k}t_v(i)$ hold. Consequently, the components $\tilde{A}_{u,k}$, $\tilde{B}_{u,k}$, and $A^l_{u,k}$ can be incrementally computed as well. Thus, the adoption within each trace can be sequentially processed and the counters can be updated accordingly. A further scan on the whole trace log enables the computation of the $A_{u,k}$ and $B_{u,k}$ components. Notice that neither the $A_{i,v,k}$ nor the $B_{i,v,k}$ need to be stored, as they are cumulated as long as the trace is processed. $\square$

Again, it is convenient to provide additional storage for the $A_{i,u,k}$ components. In fact, the term $\eta_{i,u,v,k}$ can be rewritten as $\alpha_{v,k}/A_{i,u,k}$. This relationship allows for a fast way to compute the update of $\alpha_{v,k}$ in the M step, as stated below.

Lemma 5.3. *Given a user u and a community k, let $\tilde{\alpha}_{v,k}$ be the value of $\alpha_{v,k}$ computed in the preceding iteration. Then,*

$$\alpha_{v,k} = \tilde{\alpha}_{v,k}\frac{C_{v,k} - D_{v,k}}{(T\cdot N_v - \tau_v)\Gamma_k - T\cdot\Gamma_{v,k} + E_{v,k} - F_{v,k} + \Psi_{v,k}},$$

where the counters are as defined in Table II.

Proof. As usual, we can split the computation of Equation (7), that is, $\alpha_{v,k} = num_{v,k}/den_{v,k}$. Concerning the numerator, we can then observe that

$$\begin{aligned}
num_{v,k} &= \tilde{\alpha}_{v,k}\sum_{i:v\in C_i}\sum_{u\succ_i v}\frac{\gamma_{u,k}}{A_{i,u,k}}\\
&= \tilde{\alpha}_{v,k}\left(\sum_{i:v\in C_i} C_{i,k} - \sum_{i:v\in C_i} D_{i,v,k}\right)\\
&= \tilde{\alpha}_{v,k}(C_{v,k} - D_{v,k}).
\end{aligned}$$

The remaining part can be rewritten as follows:

$$\begin{aligned}
den_{v,k} &= \sum_{i:v\in C_i}\sum_{u\notin C_i}\gamma_{u,k}(T - t_v(i)) + \sum_{i:v\in C_i}\sum_{u\succ_i v}\gamma_{u,k}(t_u(i) - t_v(i))\\
&= T\sum_{i:v\in C_i}\sum_{u\notin C_i}\gamma_{u,k} - \sum_{i:v\in C_i} t_v(i))\sum_{u\notin C_i}\gamma_{u,k}\\
&\quad + \sum_{i:v\in C_i}\sum_{u\succ_i v}\gamma_{u,k}t_u(i) - \sum_{i:v\in C_i} t_v(i)\sum_{u\succ_i v}\gamma_{u,k}\\
&= T\sum_{i:v\in C_i}(\Gamma_k - \Gamma_{i,k}) - \sum_{i:v\in C_i} t_v(i)(\Gamma_k - \Gamma_{i,k})\\
&\quad + \sum_{i:v\in C_i}(E_{i,k} - F_{i,v,k}) - \sum_{i:v\in C_i} t_v(i)(\Gamma_{i,k} - \Gamma_{i,v,k})\\
&= (TM_v - \tau_v)\Gamma_k - T\Gamma_{v,k} + E_{v,k} - F_{v,k} + \Psi_{v,k}. \quad \square
\end{aligned}$$

COROLLARY 5.4. *The complexity of the M step is $O(KL + KM + KN)$ time and space.*

PROOF. Given a pair of adjacent users in a same trace, the usual recursive relations can be observed for the $D_{i,v,k}$, $\Gamma_{i,v,k}$, and $F_{i,v,k}$ quantities. Consequently, all the components that depend on them can be computed incrementally as well, by accumulating values for each trace. Notice that the $C_{i,k}$ components require to pre-allocate the $A_{i,v,k}$ values, which can be computed in a separate scan of the whole trace in $O(KL)$ time. □

We are finally able to state the main complexity result for the C-Rate.

THEOREM 5.5. *The complexity of the EM algorithm plugged with the C-Rate model is $O(KL + KM + KN)$ in time and space.*

6. EXPERIMENTAL EVALUATION

In this section, we report an experimental analysis aimed at assessing the effectiveness of the proposed framework. Specifically, we are interested in the following aspects:

- Investigate under which conditions, and to which extent, the proposed methods can actually detect communities from propagation logs.
- Comparatively assess the adequacy of the models to fit real data, by characterizing the discovered community structures and relating them to predefined (synthetic data) and previously unknown (real-world data) structures.
- Evaluate the predictive abilities of the proposed models in terms of both activations and connections that a user is likely to exhibit.

To perform the aforementioned analysis, we rely on synthetic and real data. In both cases, we are given a set V of users, a set directed social links E between them, and a log $\mathbb{D}$ which records users' activation times on a set of propagation traces. Each link $(u, v) \in E$ represents the direction of the information flow between the two considered users, that is, information flows from u to v. Under the assumption that information can spread only by exploiting the social connections among users, the network $G = (V, E)$ will naturally shape the process of information propagation. The exact realization of users' activation times on the considered traces actually depends on set of different parameters that, without loss of generality, can be grouped into two dimensions: structural properties of the network and the choice of the propagation model, which ultimately determines how information/infections occur. This is the setting of our experimental evaluation. Even if we do not observe directly the network, we assume that

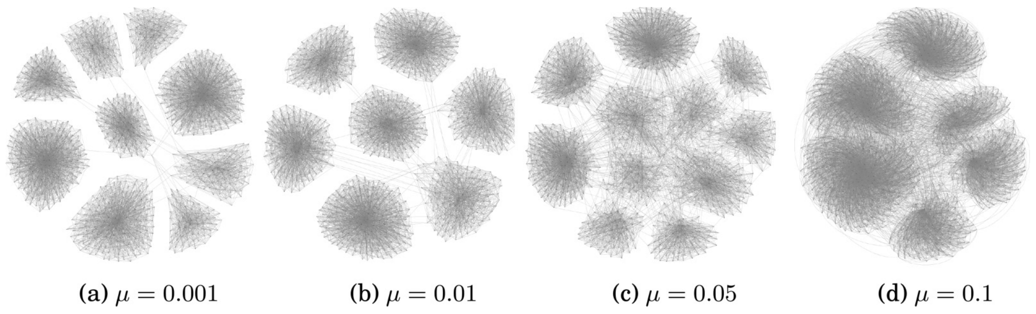

Fig. 1. Visualization of the synthetic networks generated with different values of μ.

its hidden community structure is encoded, and thus it can be inferred by analyzing the propagation log $\mathbb{D}$.

The use of synthetic data allows us to specify a ground truth, that is, a predefined community structure that we aim to discover without looking at the network. In the case of real data, we forget the network, apply our methods to detect communities by considering exclusively the propagation log, and, finally, measure the quality of the discovered communities with standard measures (e.g., conductance, cut ratio, modularity, etc.) using the network.

All experiments are run on a 2.7 GHz i7 machine by allocating a maximum of 10GB of RAM.

6.1. Synthetic Datasets

When considering synthetic datasets, the underlying assumption is that propagation traces follow the links exhibited by the underlying network. This is a well-known property of information propagation in social network, and several studies [Weng et al. 2013, 2014] witness how adoptions within a network are influenced by neighboring behavior.

Thus, in order to generate synthesized data, we proceed in two steps. First, we generate a network with a known community structure, as well as structural features typical of real networks. To this aim, we use the generator of benchmark graphs described by Lancichinetti and Fortunato [2009], which generates directed unweighted graphs with *possibly* overlapping communities. The process of network generation is controlled by the following parameters: (i) number of nodes (1,000), (ii) average in-degree (10), (iii) maximum in-degree (150), and (iv) min/max the community sizes (50/750). The four networks differ on the percentage μ of overlapping memberships, ranging into $0.001,\ 0.01,\ 0.05$, and 0.1. As is clearly visible from the topology of the generated networks reported in Figure 1, this last parameter strongly affects the structure of the network, which ranges from well-separated (but still connected) components to strongly overlapping components.

Given a network $G = (V, E)$, the next step is to generate synthetic propagation cascades by simulating a propagation/contagion process that spreads over E. In this phase, we face two main challenges: (i) limiting the bias introduced by the choice of a particular propagation model and (ii) generating propagation cascades that are likely to happen in a real-world scenario. To this purpose, we again parameterize the propagation strategy and study the behavior of each algorithm by varying such a parameter. The overall data generation schema generates $|\mathcal{I}|$ propagation traces based on the following protocol. Given a network $G = (V, E)$ with a known community structure, for each community k, an initial dummy node is connected to all nodes within the considered community, with a random influence weight sampled from the interval

Table III. Statistics for the Synthetic Data: Four Networks Corresponding to Four Values of μ as in Figure 1

	S1	S2	S3	S4
# of communities (K)	9	7	11	6
avg # of adoptions	56k	59k	82k	370k
avg length of traces	38	38	54	256
avg % of communities for trace	17%	24%	24%	82%

[0.02, 0.05]. For each trace to generate, we sample an initial dummy node, and, subsequently, we sample further dummy community nodes in sequence, where the nth node is picked randomly with probability β^n. In practice, β is a parameter controlling the number of communities from which each trace starts its diffusion. A large value of β will produce traces that start *simultaneously* in different communities. Conversely, a small value will produce traces that start in fewer communities.

At time $t = 0$, the dummy nodes determine the activation of real nodes, from which we start the subsequent diffusion process. At this stage, information can spread on the network by exploiting the links. The strength of each link is determined by considering both the out-degree ($\kappa_\cdot^{out}$) of the source and the in-degree ($\kappa_\cdot^{in}$) of the destination:

$$weight(u, v) \propto \lambda \cdot \frac{\kappa_u^{out}}{\kappa_{max}^{out}} \frac{\kappa_v^{in}}{\kappa_{max}^{in}} + (1 - \lambda) \cdot rand(0.1, 1),$$

where κ_{max}^{out} and κ_{max}^{in} are the maximum out-degree and in-degree, respectively, and λ is used to introduce a random effect. In the propagation process, the weight of each link represents a Bernoullian probability of infection. For each link, we also generate a typical infection rate $\alpha_{u,v}$, sampled from a Gamma distribution with fixed parameters (shape = 2, scale = 0.3).

To summarize, the data generation process depends on the degree of community overlapping μ, the degree of propagation overlap β, and the size $|\mathcal{I}|$ of the propagation log. In a first batch of experiments, we fix $\lambda = 0.9$, $\beta = 0.2$, and $|\mathcal{I}| = 1{,}500$ and vary the μ parameter as discussed above. For each network, we randomly generate five propagation logs. The main properties of the synthesized data are summarized in Table III. The average number of adoptions generated by the diffusion process grows with the percentage of overlapping membership, as well as the average number of communities that are involved in the propagation of each trace.

6.1.1. Baselines. The C-IC and C-Rate techniques are compared to some baseline models. The first two baselines build on the idea of network reconstruction. Given a log of past propagations $\mathbb{D}$ we can apply either NetRate or the Independent Cascade inference procedure [Saito et al. 2008], where we assume the complete graph. Both algorithms provide a set of link weights as output, and higher weights witness the existence of strong connections. We reconstruct the network by applying a sparsification procedure based on the identification of a minimum threshold value on the weights (set empirically to 10^{-14}).

Finally, communities are discovered by applying the Metis algorithm [Karypis and Kumar 1999], a scalable graph partitioning method that is reported to achieve good performances on graphs extracted from various domains. METIS is based on multilevel recursive-bisection; this allows scaling up to large-scale networks, and this is why it is often considered as a baseline method or post-processing tool in the community detection literature [Yang and Leskovec 2013; Ruan et al. 2013; Leskovec et al. 2010]. These baselines are denoted as NetRate/Metis and IC/Metis.

A further baseline is a standard clustering algorithm that groups user traces according to their likelihood of adopting the same items. Here, a user trace is represented by the set of all her/his adoptions. The method is based on a *Bernoullian expectation*

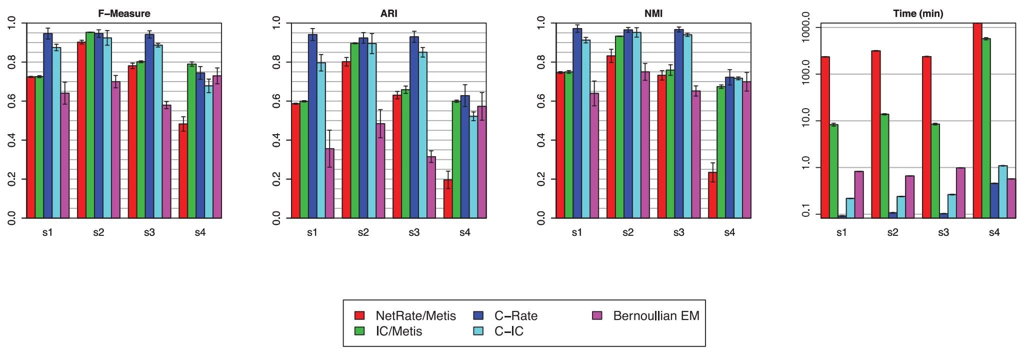

Fig. 2. Summary of the evaluation on reconstructing community structure on synthetic networks with planted communities and different levels of noise. From left to right: F-Measure, Adjusted Rand Index, Normalized Mutual Information, and runtime in minutes (log scale).

maximization procedure; the assumption is that the adoption of an item by a user can be explained in terms of a mixture of Bernoulli distributions, where each component of the mixture characterizes a community. Similar users are characterized by the same (Bernoullian) probability of adopting the same items. The output is a grouping of users according to their adoption patterns: Each community is characterized by the likelihood of adopting an item (and, consequently, to be part of a trace, with no reference to temporal information). This clustering method does not provide any information concerning the degree of influence of a user within a community.

For both the Independent Cascade and the C-IC model, we fix $\Delta = \infty$. This allows a better comparison with C-Rate, where all possible influencers are taken into account. We plan to devote the study of how the influence window affects the quality of results to a future work.

6.1.2. Results. We measure the quality of the discovered communities with respect to the known ground-truth community structure by using the Adjusted Rand Index [Jain and Dubes 1988], the F-Measure, and the Normalized Mutual Information (NMI) [Ana and Jain 2003]. For all the considered approaches, we report in Figure 2 the average quality indices, as well as their values of standard deviation relative to the five propagation logs. Both C-IC and C-Rate perform particularly well on all four networks, even if the performances degrade on the s4 network, where IC/Metis and the Bernoullian EM achieve comparable results.

While NetRate/Metis and IC/Metis show similar performances in accuracy, C-Rate exhibits higher quality than C-IC. Apparently, the approach to model community-based temporal dynamics is more effective than simply relying on time-independent reaction. In all four networks, the performances of the Bernoullian EM method are unstable, and, in general, it achieves lower-quality indices than the ones corresponding to other methods. This is a clear sign of the relevant role played by influencers when associating a user to a community: Influencers tend to better explain the activation of a user on a given propagation trace and hence tend to reduce the variability in the membership assignments.

Both C-IC and C-Rate are consistently more efficient (up to three orders of magnitude) than methods based on network reconstruction. This is due to their inference process, while the overhead introduced by the post-processing step (running Metis to detect communities) is not significant. In fact, network reconstruction methods compute pairwise influence weights by assuming a complete graph. As a consequence, they are quadratic in time and do not scale to large networks.

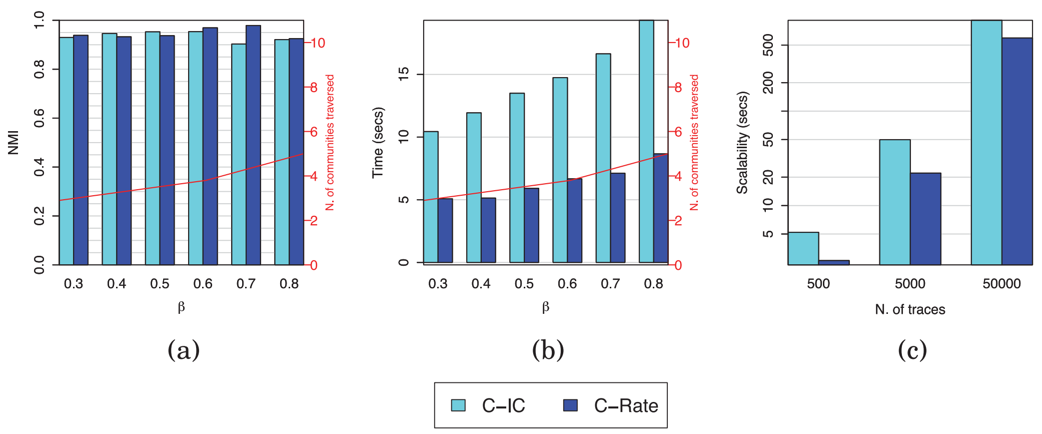

Fig. 3. Robustness at various level of the mixing parameter β: (a) and (b) show the NMI and the running time. The graph also shows the average number of communities traversed by a single trace for each dataset (red line in the graph). (c) Scalability to the number of traces.

In a second batch of experiments, we measure the effects of the mixing parameter β on C-IC and C-Rate. Higher values of β cause a trace to spread over multiple communities: As a consequence, we can measure the robustness of the algorithms by varying such a parameter. For these experiments, we use the s3 network, and we generate 1,000 traces by ranging the β within $[0.3, 0.8]$.

Figure 3(a) shows that the performances in Normalized Mutual Information do not significantly change with β: the algorithms can still separate communities and associate users with them, even when the average number of communities traversed by a single trace (denoted by the red line in the plot) increases. On the other hand, the value of the mixing parameter affects inference times, as shown in Figure 3(b). For both methods the inference phase requires more iterations to reach convergence, as a consequence of the fact that the separations between communities need to be reconstructed iteratively.

A final experiment measures the scalability of the proposed algorithms for increasing values of $|\mathcal{I}|$. In Figure 3(c) we report the running times on three log traces with increasing size, relative to the s3 network. Both algorithms scale linearly, and the general trend, where C-Rate seems more efficient than C-IC, is confirmed.

6.2. Real-World Datasets

The proposed approaches are characterized by strong assumptions on the underlying propagation model. By evaluating them on real-world data, we want to assess their flexibility and accuracy in modeling real propagation phenomena. To this purpose, we focus the next evaluation on real-world propagation traces, which have been obtained by crawling the public timeline of Twitter.[7] The canonical form of information propagation on Twitter is retweeting. However, tweets are complex entities, which can include hashtags, textual and reference information. An accurate modeling of all these entities is out of the scope of our article. Instead, we consider a simpler fragment of the propagations where we only track elementary units included in a tweet, which we identify as URLs. The assumption here is that if a tweet contains a URL, then it is more likely to be retweeted by peers sharing the same interests: Hence URL propagations can suitably highlight the underlying influence process and ultimately the underlying

[7]https://dev.twitter.com/docs/api/1/get/statuses/public_timeline.

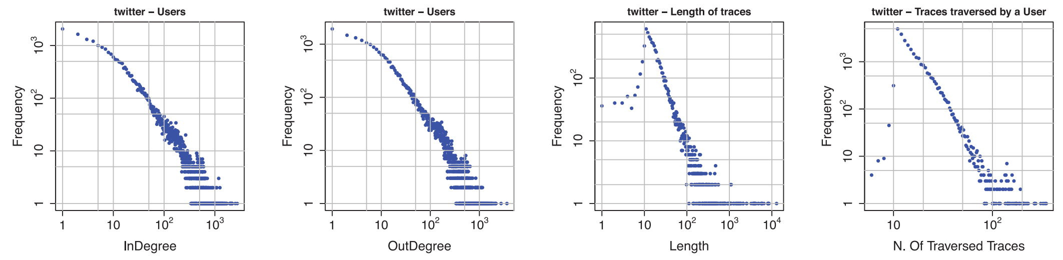

Fig. 4. Main properties of the considered real-world data. The plots show the distributions of the node degrees, trace length, and number of traces traversed by a user.

community structure. Thus, we track the propagation of URLs across the network during one month (August 2012), and each activation corresponds to the instance when a user tweets a certain URL. The raw data have been preprocessed by filtering out users who participated in fewer than five traces. Our final sample contains 28,185 nodes and 1,636,451 arcs. The activity log contains traces of 8,541 URLs for a total number of 516,412 adoptions (tweets). The average number of users per trace is 60, and, on average, a user performs 18 tweets. Other relevant features of the dataset can be observed in Figure 4. In the experiments, we only focus on traces with length greater than 10.

Since we do not have any form of ground truth for the community structure of the network under analysis, to assess the quality of the retrieved communities, we rely on empirical objective functions. For a given partition of the network, such measures promote the identification of communities that are characterized by a higher internal connectivity and that are marginally connected with the rest of the network. We consider three different scores for measuring the quality of each detected partition/communities (see Leskovec et al. [2010] for a detailed discussion):

- *Conductance* is the simplest formalization of the concept above, as it measures the ratio between the number of edges inside each considered community and the number of edges traversing its border.
- *Internal Density* measures the ratio between the actual edges and the possible edges within the community.
- *Cut Ratio* is the ratio between the number of edges on the boundary of a community and all the possible ones.

The above measures focus on evaluating the quality of each community, and they do not consider the direction of edges. We thus adopt also the directed version of *modularity* measure [Leicht and Newman 2008], which evaluates the overall quality of the partition. Modularity compares the structure of the graph to that resulting from a random graph, representing a null model, and is defined as

$$\mathcal{Q}_G = \frac{1}{m} \sum_{u,v} \left[A_{u,v} - E(u,v) \right] \delta_{c_u, c_v}.$$

In the above equation, $A_{u,v}$ is the cell of the adjacency matrix corresponding to the pair (u, v), and $E(u, v) = \kappa_u^{out} \kappa_v^{in}/m$ represents the expected likelihood of observing the link (u, v) in the null (directed random graph) model. Also, δ_{c_u,c_v} is the Kronecker delta relative to community memberships for nodes u and v.

Higher modularity, internal density and conductance, as well as lower cut ratio, denote good partitioning. In the following, we report and compare the results achieved by C-Rate, C-IC, and Bernoullian EM, as methods based on network reconstruction cannot handle this input. To obtain a reference value, we run the Metis algorithm

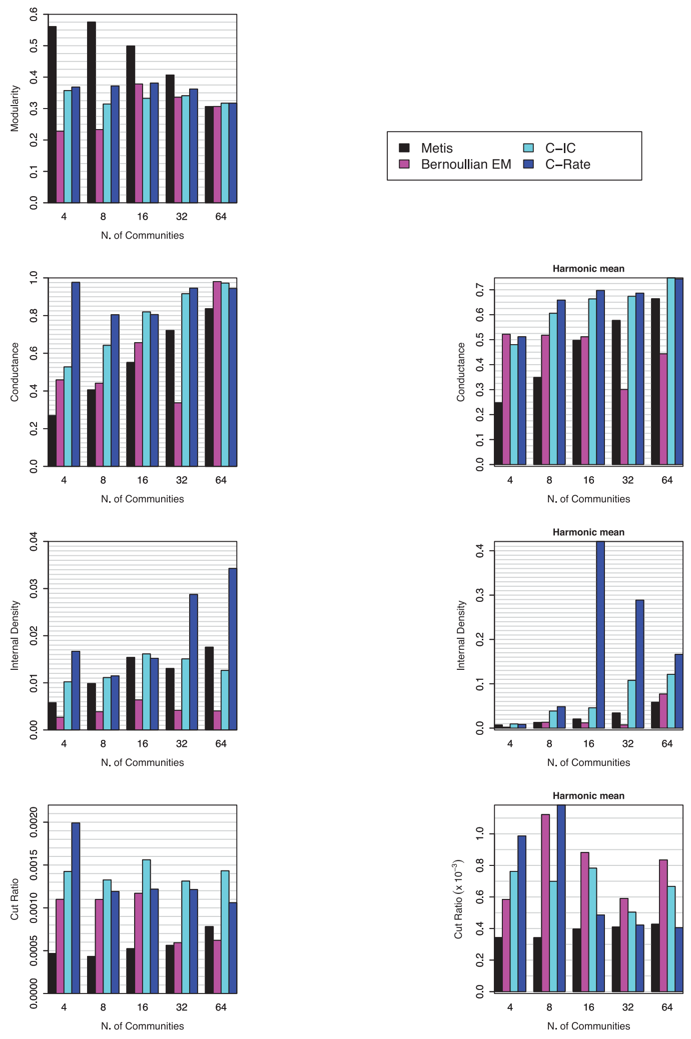

Fig. 5. Summary of the evaluation on identifying community structure on Twitter. Each row in the figure represents a measure.

to partition the network into communities and compute the quality values for such partitions. The idea here is that if the network is known, then the communities can be inferred from the links directly. By comparing this result with the results of the network oblivious community detection algorithms allows us to measure how reliable are such algorithms in detecting the underlying structure when only the propagation log is given as input.

Figure 5 reports the values of the quality measures for the aforementioned methods, and the values of conductance, internal density, and cut ratio were averaged over all the discovered communities. In this experiment, we disable the annihilation procedure and measure the quality values varying the number of communities. Surprisingly, both C-Rate and C-IC perform better than *Metis* on both conductance and internal density. C-Rate performs particularly well, even if there is no clear indication of number of communities where a method perform better than all the others on all measures. In

Table IV. Summary of the Evaluation on Twitter: C-IC and C-Rate with Annihilation

	C-IC	C-Rate
Communities	32	17
Community size (min / max / median)	113/2554/760	131/8315/1658
$\mathcal{Q}_G$	0.323	0.343
Conductance	0.794	0.687
Internal Density	0.02814	0.018
Cut Ratio	5.98×10^{-4}	4.69×10^{-4}
Time (secs)	5086	2070

particular, the modularity decreases when increasing number of communities, whereas the opposite trend can be observed for the other measures.

On the other hand, this analysis shows that both C-IC and C-Rate exhibit high values of cut ratio compared to those achieved by the Bernoullian clustering approach. In order to better analyse this situation, in Figure 5 we also plot the results on harmonic mean for those indices that are expressed as ratios. The harmonic mean in fact is known to be more appropriate when averaging ratios, as it is insensitive to outliers with high values and it tends to be dominated by low values. Thus, plotting the harmonic mean on those measures provides a better insight of the overall tendency and mitigates the effects of extremely high values. This analysis shows that

—The harmonic mean of cut ratio values exhibit now a normalized behavior, compared to the values of Figure 5. The difference in performance between the different aggregation strategy (avg. vs harmonic mean) suggests that C-IC and C-Rate tend to produce some outlier communities in which nodes have an high number of inter-community links.
—Internal density substantially increases for a large number of clusters. This is a clear sign that some communities within the partitions are densely connected and this boosts the value of harmonic mean.

To summarize, both C-IC and C-Rate exhibit good quality measures, which are comparable to, and in some cases even better than, the Metis baseline. This confirms that the analysis of the propagation traces allows us to rebuild the latent community structure with enough accuracy. Furthermore, the adoption of a propagation model based on social influence allows the identification of communities with high internal density.

Similar results can be obtained by enabling the annihilation procedure,[8] as reported in Table IV and discussed next. The resulting internal density is an order of magnitude higher than the density of the whole graph (0.0041). Also, a large number of edges tend to stay within the community they belong to, as witnessed by the values of conductance. The values of modularity are a clear sign of reliable community structure (note that acceptable modularity values are recognized starting from around 0.2–0.3). The existence of a good community structure is confirmed by the diagonal block structure in the adjacency matrices in Figure 6(a). These are adjacency matrices where the users are sorted and grouped in blocks representing the communities. Figure 6(b) represents the same adjacency matrix but with the blocks made explicit and shaded in gray according to their relative density. In both cases, the regions with higher density lie in the diagonal blocks.

Figure 6(c) summarizes influence dynamics by analyzing the distribution of the user influence weights for each community. Here, darker colors denote higher weights, and lighter ones conversely represent weights close to zero. These plots show that there is

[8]For both algorithms, we started with an initial number of 64 communities.

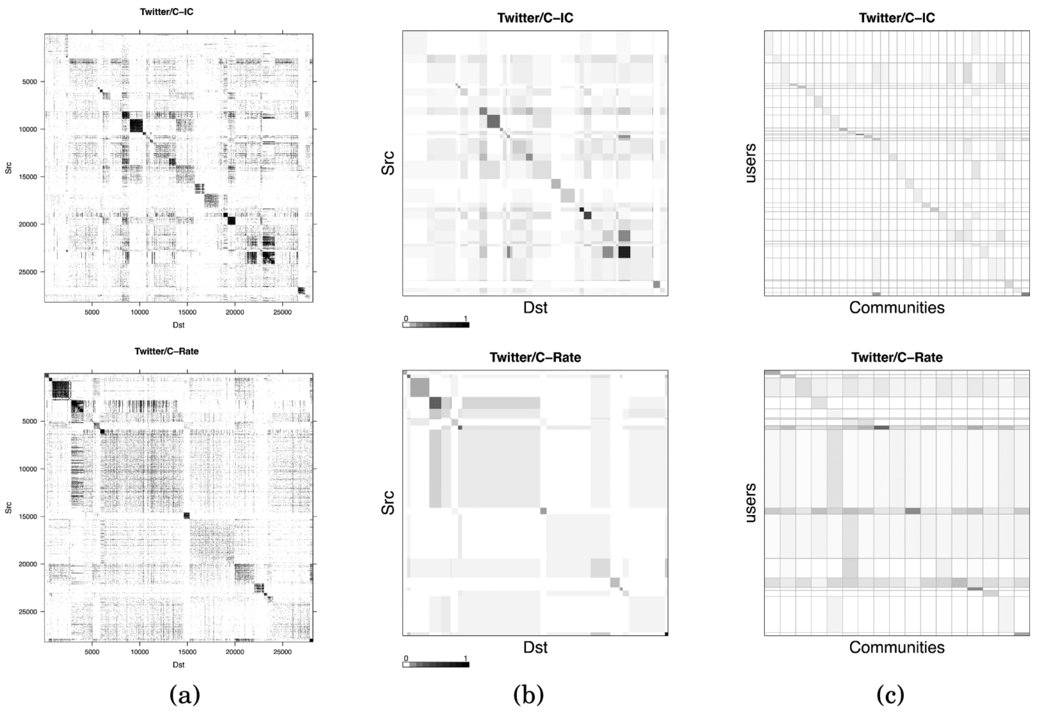

Fig. 6. (a) Adjacency block matrices for C-IC and C-Rate. (b) Density blocks within the adjacency matrix. (c) Distribution of influence weights.

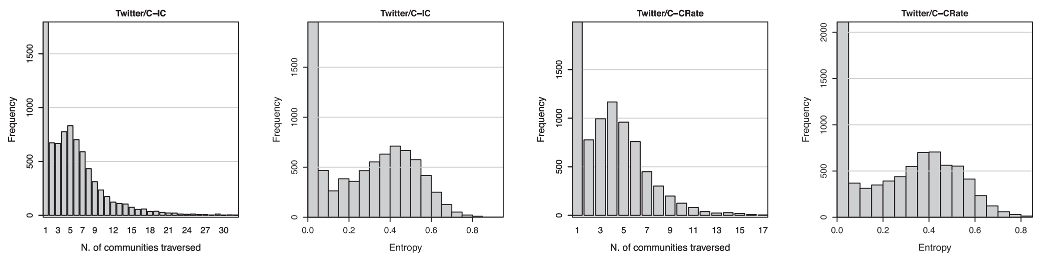

Fig. 7. Distribution of the number of communities involved in a trace.

a general tendency to express influence within the same community of membership, even if some users are able to exert influence on more communities.

Finally, Figure 7 shows the distribution of the communities involved in the diffusion of each trace. Values tend to concentrate on few communities. These plots also show the distribution of the normalized entropy of a single trace, computed by considering the frequency of each community within the trace. In practice, even when a trace touches multiple communities, low entropy values witness that there are a few predominant communities, which embody the majority of the users involved in the trace.

7. CONCLUSIONS AND FUTURE WORK

We proposed a general framework for detecting communities in a network-oblivious setting. The framework is based on the assumption that item adoptions are governed by underlying stochastic diffusion process over the unobserved social network and that such diffusion model is based on community-level influence. We instantiated the diffusion process by adopting two models that focus on both the influence exerted by a user in a given community and the likelihood of a user to belong to that community. The main difference in the two models is the way they model the contagion: C-IC is essentially a discrete-time model, whereas C-Rate models a continuous-time diffusion scenarios. The experiments show that both models are robust and effective and can be profitably employed to discover communities and regions of influence in situations where the social connections are not visible.

In this treatment, we did not cover the case where contagion can happen as the result of a cumulative effect: That is, a user can adopt an item when a significant number of users in the same community adopt that item as well. Again, two different modeling perspective can be assumed, based either on adaptations of the *Linear Threshold* model [Barbieri et al. 2013c; Kempe et al. 2003] for the case of discrete time or of the *Cox* survival model [Lee and Wang 2003] for the case of continuous time. We plan to study these models and compare them with the ones proposed in this article in our future work.

REFERENCES

L. N. F. Ana and A. K. Jain. 2003. Robust data clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 128–133.

A. Anagnostopoulos, R. Kumar, and M. Mahdian. 2008. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. 7–15.

E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. 2011. Everyone's an influencer: Quantifying influence on twitter. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM'11)*. 65–74.

N. Barbieri, F. Bonchi, and G. Manco. 2013a. Cascade-based community detection. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM'13)*. 33–42.

N. Barbieri, F. Bonchi, and G. Manco. 2013b. Influence-based network-oblivious community detection. In *Proceedings of the IEEE 13th International Conference on Data Mining (ICDM'13)*. 955–960.

N. Barbieri, F. Bonchi, and G. Manco. 2013c. Topic-aware social influence propagation models. *Knowl. Inform. Syst.* 37, 3 (2013), 555–584.

F. Bonchi. 2011. Influence propagation in social networks: A data mining perspective. *IEEE Intell. Inform. Bull.* 12, 1 (2011), 8–16.

S. Bourigault, C. Lagnier, S. Lamprier, L. Denoyer, and P. Gallinari. 2014. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM'14)*. 393–402.

W. Chen, C. Wang, and Y. Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 1029–1038.

Y. Chen, W. Zhu, W. Peng, W. Lee, and S. Lee. 2014. CIM: Community-based influence maximization in social networks. *ACM Trans. Intell. Syst. Technol.* 5, 2 (2014), 25:1–25:31.

D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. 2008. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. 160–168.

G. B. Davis and K. M. Carley. 2008. Clearing the FOG: Fuzzy, overlapping groups for social networks. *Soc. Netw.* 30, 3 (2008), 201–212.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.: Ser. B* 39 (1977), 1–38.

P. Domingos and M. Richardson. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*. 57–66.

M. A. T. Figueiredo and A. K. Jain. 2002. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 3 (2002), 381–396.

S. Fortunato. 2010. Community detection in graphs. *Phys. Rep.* 486, 3 (2010), 75–174.

M. Gomez Rodriguez, D. Balduzzi, and B. Schölkopf. 2011. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. 561–568.

M. Gomez Rodriguez, J. Leskovec, and A. Krause. 2010. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 1019–1028.

A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM'10)*. 241–250.

A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. 2011. A data-based approach to social influence maximization. *Proc. VLDB Endow.* 5, 1 (2011), 73–84.

A. K. Jain and R. C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Upper Saddle River, NJ.

G. Karypis and V. Kumar. 1999. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* 20, 1 (1999), 359–392.

D. Kempe, J. Kleinberg, and É. Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. 137–146.

M. Kimura and K. Saito. 2006. Tractable models for information diffusion in social networks. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases (PKDD'06)*. 259–271.

T. La Fond and J. Neville. 2010. Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. 601–610.

A. Lancichinetti and S. Fortunato. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* 80 (2009), 46–110.

E. T. Lee and J. W. Wang. 2003. *Statistical Methods for Survival Data Analysis*. John Wiley & Sons, Inc., Hoboken, NJ.

E. A. Leicht and M. E. J. Newman. 2008. Community structure in directed networks. *Phys. Rev. Lett.* 100, 11 (2008), 118703.

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

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. 2007b. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*. 420–429.

J. Leskovec, K. J. Lang, and M. Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. 631–640.

J. Leskovec, A. Singh, and J. Kleinberg. 2006. Patterns of influence in a recommendation network. In *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06)*. 380–389.

M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. 2011. Sparsification of influence networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. 529–537.

Y. Mehmood, N. Barbieri, F. Bonchi, and A. Ukkonen. 2013. CSI: Community-level social influence analysis. In *Proceedings o the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD'13), Part II*. 48–63.

M. E. J. Newman and E. A. Leicht. 2007. Mixture models and exploratory analysis in networks. *Proc. Natl. Acad. Sci. U.S.A.* 104, 23 (2007), 9564–9569.

W. Ren, G. Yan, X. Liao, and L. Xiao. 2009. Simple probabilistic algorithm for detecting community structure. *Phys. Rev. E* 79, 3 (2009), 36–111.

D. M. Romero, B. Meeder, and J. Kleinberg. 2011. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*. 695–704.

Y. Ruan, D. Fuhry, and S. Parthasarathy. 2013. Efficient community detection in large networks using content and links. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. 1089–1098.

K. Saito, R. Nakano, and M. Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Part III (KES'08)*. 67–75.

H. Shawndra, P. Foster, and V. Chris. 2006. Network-based marketing: Identifying likely adopters via consumer networks. *Stat. Sci.* 21, 2 (2006), 256–276.

Y. Wang, G. Cong, G. Song, and K. Xie. 2010. Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. 1039–1048.

Y. Wang, H. Shen, S. Liu, and X. Cheng. 2015. Learning user-specific latent influence and susceptibility from information cascades. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. 477–483.

L. Weng, F. Menczer, and Y. Ahn. 2013. Virality prediction and community structure in social networks. *Sci. Rep.* 3 (2013).

L. Weng, F. Menczer, and Y. Ahn. 2014. Predicting successful memes using network and community structure. In *Proceedings of the 8th International Conference on Weblogs and Social Media (KDDICWSM'14)*.

L. Weng, J. Ratkiewicz, N. Perra, B. Gonçalves, C. Castillo, F. Bonchi, R. Schifanella, F. Menczer, and A. Flammini. 2013. The role of information diffusion in the evolution of social networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. 356–364.

J. Yang and J. Leskovec. 2013. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM'13)*. 587–596.

Received May 2015; revised March 2016; accepted July 2016