Cascade-based Community Detection

Nicola Barbieri
Yahoo! Research
Barcelona, Spain
barbieri@yahoo-inc.com

Francesco Bonchi
Yahoo! Research
Barcelona, Spain
bonchi@yahoo-inc.com

Giuseppe Manco
ICAR-CNR
Rende, Italy
manco@icar.cnr.it

ABSTRACT

Given a directed social graph and a set of past information cascades observed over the graph, we study the novel problem of detecting modules of the graph (communities of nodes), *that also explain the cascades.* Our key observation is that both information propagation and social ties formation in a social network can be explained according to the same latent factor, which ultimately guide a user behavior within the network. Based on this observation, we 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 observed set of cascades. Our model produces overlapping communities and for each node, its level of authority and passive interest in each community it belongs.

For learning the parameters of the CCN model, we devise a Generalized Expectation Maximization procedure. We then apply our model to real-world social networks and information cascades: the results witness the validity of the proposed CCN model, providing useful insights on its significance for analyzing social behavior.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Social networks, Community Detection, Information Cascades

1. INTRODUCTION

Understanding how the adoption of new practices, ideas, beliefs, technologies and products can spread trough a population driven by *social influence*, is a central issue for the whole of social sciences. Taking into account the modular structure of the underlying social network provides further important insight in the phenomena known as *social contagion* or *information cascades.* In particular, individuals tend to adopt the behavior of their social peers, so that cascades happen first locally, within close-knit communities, and become global “viral” phenomena only when they are able cross the boundaries of these densely connected clusters of people. Therefore, the study of social contagion is intrinsically connected to the problem of understanding the modular structure of networks (known as *community detection*), and together form the central core of network science.

Recently with the explosion of on-line social platforms such as `Twitter` and `Facebook`, the interest in these topics has exploded accordingly. Researchers have investigated how to exploit social influence for “word-of-mouth” driven viral marketing applications [17, 44, 28, 10], studying person-to-person recommendation for purchasing books and videos [36, 35] or telecommunications services [27] finding conditions under which such recommendations are successful. Others have focussed on the important problems of how to measure social influence [47, 50, 24, 53] and how to distinguishing real social influence from “homophily” and other external factors of correlation [3, 16, 4, 20]. Finally, a growing effort has been devoted to the analysis of influence-driven information cascades in `Twitter` [14, 52, 5, 45].

At the same time a large effort has been devoted to develop community detection algorithms (surveyed later in Section 1.2), but quite surprisingly there has not been much effort to explain the modular structure of social networks and the phenomenon of social contagion, *jointly.* We say that the lack of research in the intersection of these two themes is surprising because, as discussed above, they are intrinsically connected. This is expressed nicely by Easley and Kleinberg in their book [18, page 577]:

> *“cascades and clusters truly are natural opposites: clusters block the spread of cascades, and whenever a cascade comes to a stop, there’s a cluster that can be used to explain why.”*

If a cluster can explain why a cascade comes to a stop, then observing past cascades we can find out something about the existence of clusters. Inspired by this observation, we propose to take benefit of an available set of traces of past information cascades, in order to better determine the community structure of the underlying social network.

1.1 Our proposal

Given a graph and a set of cascades, we tackle the community detection task by fitting a unique stochastic generative model to the observed social graph and cascades.

Since we deal with information propagation, it is natural to consider a *directed* social graph $G = (N, A)$ where an

WSDM’13, February 4–8, 2012, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$15.00.

arc $(u, v) \in A$ indicates that v is a follower of u, i.e., v is notified about u's activity. In other terms, information can propagate from u to v. For instance, in Twitter v can see something posted by u and re-tweet so that the information becomes available to her own followers, thus propagating further in the network. A cascade of an item i is a sequence of pairs $(user, timestamp)$ recording which nodes adopt i (e.g., re-tweeting it, in the Twitter example) and at which time. We are given a set of such propagations.

In our CCN model we assume that each observation is the result of a stochastic process where a given user u acts in the network according to a set of topics/communities which also represent her interests. Given a community c, the degree of involvement of user u to that community is governed by two parameters, namely $\pi_u^{c,s}$ and $\pi_u^{c,d}$. Specifically, $\pi_u^{c,s}$ measures the degree of "active involvement" of u in c, while $\pi_u^{c,d}$ measures the degree of "passive involvement" of u in c.

Let us use the Twitter example again and let us consider an hypothetical user u which uses the microblogging platform for three specific interests: (i) network science and data mining, (ii) the city of Barcelona, and (iii) the rock legend Bruce Springsteen. While on the first topic u is actively posting, using the platform for communicating with other researchers, in the other two topics u is just passively listening: for sake of information needs, u follows users which are good information sources for events happening in Barcelona and users which are authorities for whatever concerns Bruce Springsteen. These users which are good sources of information usually have a large number of followers and are, in some sense, "influential". In the second and third community u might re-tweet some pieces of information, but it is quite unlikely that u would produce some original information.

Going back to our parameters we can expect u to have both high $\pi_u^{c,s}$ and $\pi_u^{c,d}$ in the first community. Instead, in the other two communities we can expect u to have a low $\pi_u^{c,s}$ and a high $\pi_u^{c,d}$. Not surprisingly u has many followers in the first community and almost no followers in the other two communities.

Indeed, this is the key observation behind our model: the likelihood of u posting something on a topic, the likelihood of this information being further propagated, and the likelihood of u having followers interested in that topic, are all strongly correlated. In our model they are jointly represented by the parameter $\pi_u^{c,s}$. Similarly, we model the likelihood of having an incoming arc in a community and the likelihood of being influenced by other users in that community with the parameter $\pi_u^{c,d}$. This is how we achieve the jointly modeling of the social graph and the set of cascades.

Another important observation is in place. As seen in the Twitter example one user can belong to more than one community, but a link is usually explainable because of a unique topic. In graph terminology this translates in the fact that the communities of nodes are *overlapping* and induced by a *partitioning of the links*. In our model information starts in a community, propagates trough the links of that community, and can jump in another community only thanks to users which are in the overlap of the two communities.

1.2 Related work

Communities have a long history in social sciences, but it was in 2002 that the seminal paper by Girvan and Newman [23] triggered a lot of interest on the problem of *community detection*, which has since then been extensively studied, mainly in the physics and in the computer science literature. Until recently most of such literature[1], has focussed on finding disjoint communities in simple undirected graphs. That is to say that (i) nodes can belong to one and only one cluster, (ii) the relation between the nodes is symmetric, and (iii) no additional information is considered beyond the graph structure. In the following we review various proposals that have dropped one or more of these assumptions.

Overlapping communities. Nowadays is widely understood and accepted that people in social networks rarely belong to only one community: for instance the same individual usually has family, friends, colleagues and several interest-based affiliations. This idea has also been explicitly implemented in Google+ "circles" or Facebook "smart lists".

A recent survey by Xie et al. [54] categorizes algorithms for overlapping community detection in various classes: methods based on *clique percolation* [41, 42, 30]; methods that extend the idea of *label propagation* [43] to produce overlapping communities [25, 56, 55]; agent-based and particles-based models [15, 12]; methods based on local expansion and optimization [7, 31, 32, 40]. But the two classes more relevant for our proposal are *link partitioning* methods and *stochastic generative* models, discussed next.

Link partitioning has recently gained popularity [1, 19]. Clustering links instead of nodes is a very appealing approach to obtain overlapping communities: it is simple, more understandable and realistic than simply having a soft (or fuzzy) assignment of nodes to communities. In fact, as highlighted before, while for nodes is natural to belong to more than a community, links are usually explainable by a co-affiliation to some topic/community (as in the famous *affiliation networks* [33] model). Evans and Lambiotte [19] use the idea of applying normal node partitioning to the *line graph* of the given network, in order to obtain a link partitioning in the original network. Ahn et al. [1] use a simple hierarchical clustering of the links, where the similarity among two links incident in the same node is defined based on the Jaccard coefficient of the neighborhoods of the other two nodes. Kim and Jeong [29] extend the *Infomap* method to the line graph which encodes the path of the random walk on the line network under MDL.

While these methods are based on heuristic quality function, in recent years approaches based on fitting a generative model to the data have emerged. Airoldi et al. [2] introduce *mixed membership block model*: this technique factorizes the adjacency matrix in a low dimensional space expressing patterns of directed social relationships between blocks of vertices. According to the generative process, for each pair of nodes group membership is sampled for both for the source and the destination: the link is generated by sampling from the binomial distribution which encodes the probability of observing a directed connection between the considered groups. Since membership assignment are drawn independently for each possible link, users can belong to multiple groups.

A different generative model is proposed by Ball et al. [6]. The basic assumption is the existence of $n \times k$ parameters, where n is the number of nodes and k the number of communities, which specify the propensity of each vertex to have links of each possible label. The number of links of label z between two vertices is then assumed to be distributed

[1] See Fortunato [21] for a comprehensive survey.

according to a Poisson distribution parameterized by the product of the two vertex-label components.

Despite their elegant formulation, both [6, 2] suffer from a main drawback: their parameter estimation phase requires to estimate the link probability between any pair of vertices and this might be unsuitable for large networks.

Communities in directed graphs. Regardless the wide availability of directed networks (e.g., Twitter or the WWW itself), methods for community detection in directed networks are relatively new: for many years the most common practice to deal with directed networks has been to ignore directionality and apply the methods developed for undirected networks. In our setting, as we want to study communities and information propagation jointly, ignoring directionality would be an unacceptable loss of relevant information.

One of the first proposal is by Guimerà et al. [26] consider a bipartite graph with *actors* on one side and *teams* on the other, and they propose to optimize a measure of *bipartite modularity*, which adapts modularity to the bipartite case. Another recent modularity optimization approach is by Leicht and Newman [34]. Neither of the two approaches above consider overlapping communities. Palla et al. [42] extend the method of *clique percolation* [41] to deal with directed networks by considering *directed k-cliques*, which are complete sub-graphs of size k in which an ordering can be made such that between any pair of nodes there is a directed link from the higher order node towards the lower one. This work and [2] are among the few methods we are aware of, that can produce overlapping communities from directed networks.

When additional information is available. Most of the literature on community detection focuses on finding (either disjoint or overlapping) groups of nodes from a given (either undirected or directed) graph. However, in online social networking sites, richer data is available. Beyond the mere link information, users might be annotated with, e.g., demographical information, shopping behavior, interests, tags and so on. Also the links might be labeled with a relationship type: e.g., family, friend, colleague and so on. Similarly nodes and links in biological networks are typically labeled with additional information. Recently researchers (mainly in the data mining community) have started proposing method to discover communities in nodes-attributed graphs [22, 37, 57, 49] and in edge-labeled graphs [11].

The two papers which are probably the most related to ours are [46, 51]. Sachan al. [46] use topic-detection methods over the content of exchanged messages for discovering overlapping communities of nodes. The main distinction with our proposal is that we focus on the structure of the propagations without modelling the content. Wang et al. [51] study the *influence maximization* problem [28] for finding the Top-K influential nodes in *mobile* social networks. In order to speed-up the costly influence maximization process, they propose to first detect communities and then assume that the influence of a user is limited to her community. Therefore they propose an algorithm based on label propagation [43] where the propagation follows the *independent cascade* model [28]. In other terms, they modify label propagation to deal with a directed graph whose links have associated an *influence probability.* This is different from modeling a social graph and a set of cascades jointly, as we do in this paper. Another difference is that [51] produces disjoint communities.

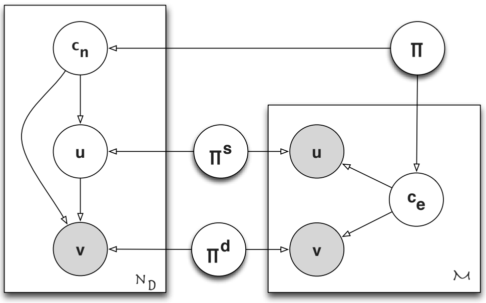

Figure 1: The CCN generative model.

2. THE CCN MODEL

We are given a directed graph $G = (N, A)$, where N is the set of the nodes and a link $(u, v) \in A$ indicates that v is a follower of u, i.e., v is notified about u's activities. Let n and m be the cardinality of N and A respectively.

We are also given a log $\mathbb{L}$ of past information cascades defined on a set of items $\mathcal{I}$ and on the nodes of the network. A tuple $(i, u, t) \in \mathbb{L}$ represents the fact that user u adopted i at time t. Let $t_i(u)$ be the activation timestamp for the user u on i; we assume that no user adopts the same item more than once and $t_i(u) = \infty$ if u does not adopt i in $\mathbb{L}$. We also denote by $D_i(t)$ the set of users that become active on i at time t, and $C_i(t)$ will denote the set of user who are active on i by time t, i.e $C_i(t) = \bigcup_{t' \leq t} D_i(t')$.

When a node u adopts i, we can think of this as information flowing from u's neighbors that have adopted i before. We assume that only users that adopted i *recently* (i.e., within an *influence window* represented by a threshold Δ) can influence their neighbors. Let t represent the current timestamp. We define the set $\mathcal{F}_i(t)$ of the nodes that can propagate i at the current time:

$$\mathcal{F}_i(t) = \{u \in N | t - t_i(u) \leq \Delta\}.$$

We also define the set of u's neighbors, $\mathcal{F}_{i,u}(t)$ that potentially can have influenced u in the selection of i at time t:

$$\mathcal{F}_{i,u} = \{v \in N | (v, u) \in A \wedge 0 \leq t_i(u) - t_i(v) \leq \Delta\}.$$

Since we are interested in modeling only activation which are likely to be triggered by some influencer (as the others represent autonomous activations), we will focus on the subset $\mathbb{D}$ of $\mathbb{L}$ defined as $\mathbb{D} = \{(i, v, t) \in \mathbb{L} : \mathcal{F}_{i,v} \neq \emptyset\}$. Let n_d denote the cardinality of $\mathbb{D}$.

As justified in Section 1.1 we assume that each observation is the result of a stochastic process where a given user u acts in the network according to a set of topics/communities which also represent her interests. Given a community c, the degree of involvement of user u to that community is governed by two parameters, namely $\pi_u^{c,s}$ and $\pi_u^{c,d}$. Specifically, $\pi_u^{c,s}$ measures the degree of "active involvement" of u in c, while $\pi_u^{k,d}$ measures the degree of "passive involvement" of u in c.

Figure 1 provides a description of the dependencies in the generative process. There are 3 prior components representing, respectively:

- the probability (Π) to observe a phenomenon in a certain topic/community,
- the level of active Π^s and passive Π^d interest of each user in each community.

Each phenomenon in the social network can be explained according to the above priors:

- a link (u, v) can only be observed if, by picking a latent topic/community c, u has a high degree of activeness and v have a high degree of passive interest in c.
- Analogously, a user v can only perform an action i if there is a latent topic/community k that likely enables an influencer u, and v can be influenced by u according to their degrees of activeness and subordination.

An interesting perspective is as follows. The action log $\mathbb{D}$ can be interpreted as a weighted bipartite graph, where links are only possible between the set of items $\mathcal{I}$ and the set of users N. The weight for a link is given by the timestamp $t_i(u)$. Under this perspective, a link (u, v) in G is only possible if a latent factor k makes it possible that u is the source and v is the destination. Analogously, a link (i, v) in $\mathbb{D}$ is only possible if there are *two* latent factors: namely, a community k and an influencer u, such that u has a high influence ratio, and v a high subordination ratio.

Based on these premises, a social network graph G and the set of observed diffusions $\mathbb{D}$ are drawn according to the following generative model:

For each link $\ell = (u, v) \in A$:

- sample a community $c_\ell \sim \mathit{Discrete}(\Pi)$
- sample a source $u \sim \mathit{Discrete}(\vec{\vartheta}^{c_\ell})$
- sample a destination $v \sim \mathit{Discrete}(\vec{\varphi}^{c_\ell})$

For each item $i \in \mathcal{I}$ and timestamp t

- Sample the number of activations $a_{i,t}$
- for each activation a where $1 \leq a \leq a_{i,t}$
  - Sample a community $c_a \sim \mathit{Discrete}\,(\Pi)$
  - pick an influencer $u_a \sim \mathit{Discrete}\left(\vec{\theta}_u^{c_a,a}\right)$
  - sample activation $v_a \sim \mathit{Discrete}\left(\vec{\phi}_u^{c_a,a}\right)$

The scheme assumes that there are two independent processes, regarding respectively the generation of the directed links and of the activations. For the first process, source and destination of a link are based on two multinomial distributions $\vec{\vartheta}^k$ and $\vec{\varphi}^k$. In our modeling we assume that both these parameters are defined in terms of the prior parameters $\vec{\pi}^{k,s}$ and $\vec{\pi}^{k,d}$, as follows:

$$\vartheta_u^k = \frac{\exp\left\{\pi_u^{k,s}\right\}}{\sum_{\overline{u}\in N} \exp\left\{\pi_{\overline{u}}^{k,s}\right\}} \tag{1}$$

$$\varphi_u^k = \frac{\exp\left\{\pi_u^{k,d}\right\}}{\sum_{\overline{u}\in N} \exp\left\{\pi_{\overline{u}}^{k,d}\right\}} \tag{2}$$

There is an interesting correlation between the above formulas and the traditional *Preferential Attachment* model: if we interpret the terms $\exp\left\{\pi_u^{k,s}\right\} / \exp\left\{\pi_u^{k,d}\right\}$ as the indegree/outdegree of node u, then the likelihood of u to be connected is compatible with the above mentioned model, in a community-based fashion.

The second process takes a pair item/timestamp, and delves into a sequence of activations. In practice, at each timestamp, we sample the activations for the item under consideration. Since an activation a involves both an influencer and an influenced, we pick them from two multinomial distributions $\vec{\theta}^{k,a}$ and $\vec{\phi}_u^{c_a,a}$. With an abuse of notation, we associate with each activation a a timestamp t_a and an action i_a (besides the active user v_a and the latent influencer u_a). Then, we can define the multinomials as follows:

$$\theta_u^{k,a} = \frac{\exp\left\{\pi_u^{k,s}\right\}}{\sum_{u'\in\mathcal{F}_{i_a}(t_a)} \exp\left\{\pi_{u'}^{k,s}\right\}} \tag{3}$$

$$\phi_{u,v}^{k,a} = \frac{\exp\left\{\pi_v^{k,d}\right\}}{\sum_{v':(u,v')\in A, v'\notin C_{i_a}(t_a-1)} \exp\left\{\pi_{v'}^{k,d}\right\}} \tag{4}$$

Notice that an influencer u is chosen among those users who can propagate i, that is, $u \in \mathcal{F}_{i_a}(t_a)$. Analogously, a node v can only be influenced to adopt i_a if it is connected to the influencer, and if it is still inactive: $(u, v) \in A$ and $v \notin C_{i_a}(t_a - 1)$. Again, by looking at the interpretation of the action log as a bipartite graph, a new link connects i to a user v, according to a markovian process that selects an influencer u among those who already expressed a links with i in the past, and then by choosing v among those connected to u.

The above multinomials are both conditioned to the hyperparameters Π^s and Π^d, where $\Pi^s = \{\vec{\pi}^{1,s}, \ldots, \vec{\pi}^{K,s}\}$ and $\Pi^d = \{\vec{\pi}^{1,d}, \ldots, \vec{\pi}^{K,d}\}$, which associate to each user its degree of active and passive interest within a given community. Finally, $\Pi = \{\pi_1, \ldots, \pi_K\}$ represents the prior multinomial distribution of the K communities involved in the process.

3. LEARNING

Given the social graph G, the propagations log $\mathbb{D}$ and a positive integer K, our aim is to detect and estimate a set of K overlapping communities, which are specified by the parameter set $\Theta = \{\Pi, \Pi^s, \Pi^d\}$. Assuming links and propagation traces independent, the likelihood of the data given the model parameters Θ, can be expressed as:

$$P(G, \mathbb{D}|\Theta) = \prod_{(u,v)\in A} P(u, v|\Theta) \cdot \prod_{a\in\mathbb{D}} P(a|\Theta) \tag{7}$$

where

$$P(u, v|\Theta) = \sum_k \vartheta_u^k \varphi_v^k \pi_k$$

and

$$P(a|\Theta) = \sum_k \sum_{u\in\mathcal{F}_{i_a,v_a}} \pi_k \theta_u^{k,a} \phi_{u,v_a}^{k,a}$$

The above likelihood is difficult to maximize, due to the presence of three latent variables modeling respectively the community assignments and the latent influencer. Making these variables explicit through binary assignments z_ℓ^k, z_a^k and w_a^u, yields:

$$\begin{aligned} P(G, \mathbb{D}, Z_G, Z_{\mathbb{D}}, W_{\mathbb{D}}|\Theta) = &\prod_{\ell\equiv(u,v)\in A} \prod_k \left(\vartheta_u^k \varphi_v^k \pi_k\right)^{z_\ell^k} \\ &\cdot \prod_{a\in\mathbb{D}} \prod_k \prod_{u\in\mathcal{F}_{i_a,v_a}} \left(\pi_k \theta_u^{k,a} \phi_{u,v_a}^{k,a}\right)^{z_a^k w_a^u} \end{aligned} \tag{8}$$

$$\delta_{\bar{u}}^k = \log \left\{ \frac{\sum_{v:(\bar{u},v)\in A} \gamma_{\bar{u},v,k} + \sum_{a\in\mathbb{D}:\bar{u}\in\mathcal{F}_{i_a,v_a}} \eta_{\bar{u},a,k}}{\vartheta_{\bar{u}}^k \sum_{(u,v)\in A} \gamma_{u,v,k} + \sum_{a\in\mathbb{D}:\bar{u}\in\mathcal{F}_{i_a}(t_a)} \theta_{\bar{u}}^{k,a} \sum_{u\in\mathcal{F}_{i_a,v_a}} \eta_{u,a,k}} \right\} \tag{5}$$

$$\lambda_{\bar{v}}^k = \log \left\{ \frac{\sum_{u:(u,\bar{v})\in A} \gamma_{u,\bar{v},k} + \sum_{a:v_a=\bar{v}} \sum_{u\in\mathcal{F}_{i_a,\bar{v}}} \eta_{u,a,k}}{\varphi_{\bar{v}}^k \sum_{(u,v)\in A} \gamma_{u,v,k} + \sum_{a:\bar{v}\notin C_{i_a}(t_a-1)} \sum_{u\in\mathcal{F}_{i_a,v_a}:(u,\bar{v})\in A} \eta_{u,a,k}\phi_{u,\bar{v}}^{k,a}} \right\} \tag{6}$$

Figure 2: Update formulas for the GEM procedure.

As a consequence, the expected complete-data log-likelihood can be defined as:

$$Q(\Theta,\Theta') = \sum_{(u,v)\in A} \sum_k \gamma_{u,v,k}(\Theta') \left[\log \pi_k + \log \vartheta_u^k + \log \varphi_v^k\right]$$
$$+ \sum_{a\in\mathbb{D}} \sum_k \sum_{u\in\mathcal{F}_{i_a,v_a}} \eta_{u,a,k}(\Theta') \left(\log \pi_k + \log \theta_u^{k,a} + \log \phi_{u,v_a}^{k,a}\right)$$

where

$$\begin{aligned}\gamma_{u,v,k}(\Theta) &= P(z_\ell^k | \ell \equiv (u,v) \in A, \Theta)\\ &= \frac{\vartheta_u^k \varphi_v^k \pi_k}{\sum_{k'} \vartheta_u^{k'} \varphi_v^{k'} \pi_{k'}}\end{aligned} \tag{9}$$

and

$$\begin{aligned}\eta_{u,a,k}(\Theta) &= P(z_a^k, w_a^u | a \in \mathbb{D}, \Theta)\\ &= \frac{P(a\in\mathbb{D}|w_a^u, z_a^k, \Theta) P(w_a^u|z_a^k,\Theta) P(z_a^k|\Theta)}{P(a\in\mathbb{D}|\Theta)}\\ &= \frac{\phi_{u,v_a}^{k,a} \theta_u^{k,a} \pi_k}{\sum_{k'} \sum_{u'\in\mathcal{F}_{i_a,v_a}} \phi_{u',v_a}^{k',a} \theta_{u'}^{k',a} \pi_{k'}}\end{aligned} \tag{10}$$

The standard EM approach expects that, given a previous value Θ^{old}, $Q(\Theta,\Theta^{old})$ can be optimized with respect to Θ. An optimal solutions for Π is straightforward:

$$\pi_k = \frac{\sum_{(u,v)\in A} \gamma_{u,v,k} + \sum_{a\in\mathbb{D}} \sum_{u\in\mathcal{F}_{i_a,v_a}} \eta_{u,a,k}}{m + \sum_{a\in\mathbb{D}} \sum_k \sum_{u\in\mathcal{F}_{i_a,v_a}} \eta_{u,a,k}} \tag{11}$$

The remaining parameters cannot be solved in a closed form, due essentially to the non-linearity of the multinomial parameters. We overcome this limitation by resorting to a slight modification of the EM approach, which combines the *Improved Iterative Scaling* algorithm in the *Generalized Expectation-Maximization (GEM)* procedure [38]. Rather than maximizing $Q(\Theta,\Theta^{old})$, we look for an upgrade Γ of Θ^{old} that guarantees

$$Q(\Theta^{old} + \Gamma, \Theta^{old}) \geq Q(\Theta^{old}, \Theta^{old})$$

We introduce for each $\pi_u^{k,s}$ an upgrade δ_u^k, and for each $\pi_v^{k,d}$ an upgrade λ_u^k. By algebraic manipulations, the above inequality yields the updates 5 and 6 in Figure 2. It is worth noticing how both the updates add two main contributions, coming respectively from the links in the graph and the propagations log.

The general scheme of the parameter estimation is given in Algorithm 1. The initialization can be accomplished in several different ways, for instance: (i) randomly, enforcing the constraint $\sum_k \pi_k = 1$ and with $\pi_u^s, \pi_u^d \in [-\pi_{MIN}, \pi_{MAX}]$ (e.g., $[-2,2]$); (ii) by fitting a simpler model (e.g., considering only the social graph structure). Also, the estimation of the number of communities K is accomplished through model selection, by resorting to the standard BIC criterion [48] which penalizes the log likelihood in a way proportional to the complexity of the model:

Algorithm 1: CCN model parameters estimation

```
Input  : Graph $G = (N, A)$, propagation log $\mathbb{D}$, and
         $K \in \mathbb{N}^+$.
Output: The set of all parameters, $\Theta = \{\Pi, \Pi^s, \Pi^d\}$ .

$init(\Pi, \Pi^s, \Pi^d)$; //Initialization of parameters
repeat
    E-step
    forall the $k \in \{1, \dots K\}$ do
        forall the $\ell \equiv (u, v) \in A$ do
            compute $\gamma_{u,v,k}$ according to eq. 9
        end
        forall the $a \equiv (i_a, v_a, t_a) \in \mathbb{D}$ do
            forall the $u \in N$ do
                compute $\eta_{u,a,k}$ according to eq. 10
            end
        end
    end
    M-step
    forall the $k \in \{1, \dots K\}$ do
        compute $\pi_k$ according to equation 11
        forall the $u \in N$ do
            compute $\delta_u^k$ and $\lambda_u^k$ according to eqs. 5-6
            $\pi_u^{k,s(new)} \leftarrow \pi_u^{k,s} + \delta_u^k$
            $\pi_u^{k,d(new)} \leftarrow \pi_u^{k,d} + \lambda_u^k$
        end
    end
until convergence;
```

$$BIC(\Theta) = -2 \log P(G, \mathbb{D}|\Theta) + C \log(m + n_d) \tag{12}$$

where C represents the number of free parameters, that is, $C = |\Pi| - 1 + |\Pi^s| + |\Pi^d| = K(2n+1) - 1$.

4. EXPERIMENTAL EVALUATION

The joint modeling of the incidence matrix and the propagation logs, provides a powerful framework to detect and understand different patterns within the data. In this section we analyze the application of the proposed technique to real networks and propagation logs. More specifically, we are interested in:

- investigating whether the CCN method is actually capable of detecting communities within data;
- assessing the adequacy of the model to fit real data;
- unveiling the complex and hidden relationships between groups and influence propagation;
- charactering the discovered community structures.

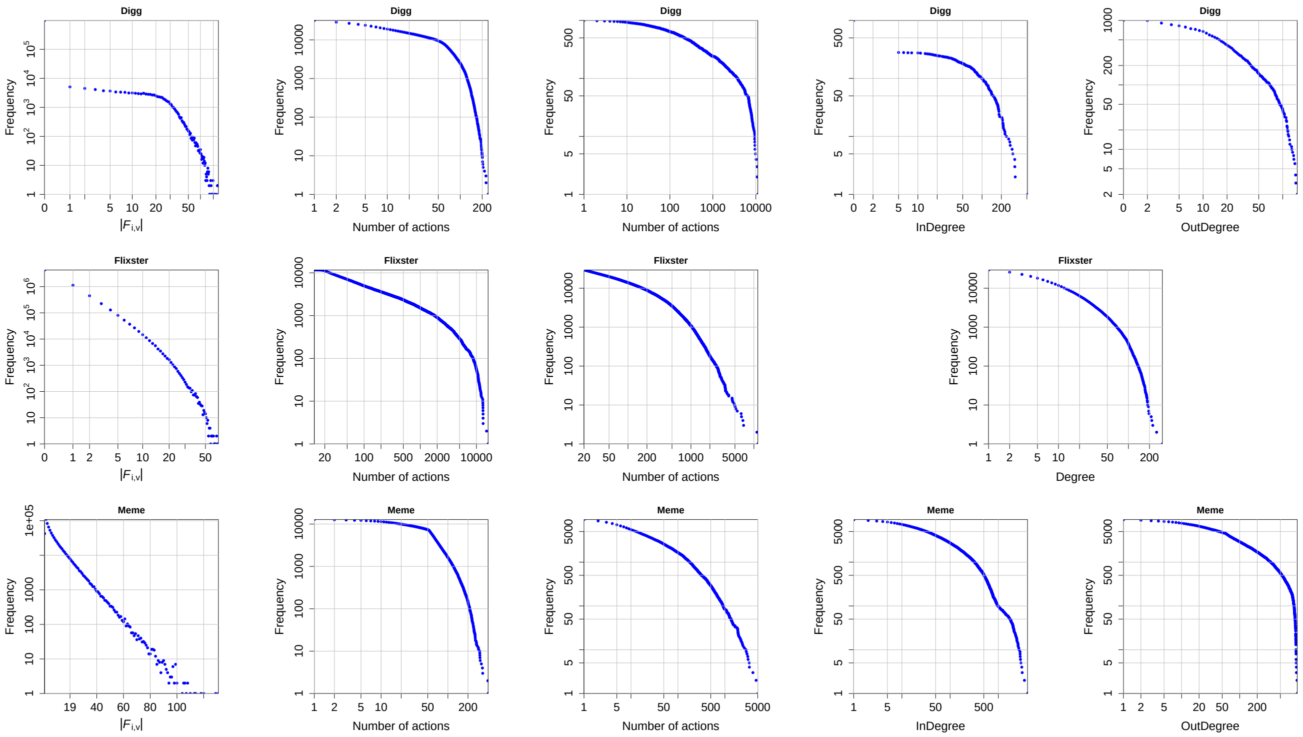

Figure 3: Cumulative distribution of $\mathcal{F}_{i,v}$ (col. 1); cumulative distribution of actions per item (col. 2); cumulative distribution of actions per user (col. 3); cumulative InDegree (col. 4); cumulative OutDegree (col. 5).

Datasets. We use three real-world datasets, each of them containing both a social graph G and a log $\mathbb{D}$ of past activations relative to a given time interval. These datasets come from Digg[2] (`www.digg.com`), Flixster [3] (`www.flixster.com`) and Yahoo! Meme (`http://meme.yahoo.com`). Digg is a social news website, where users vote stories. In this case the activation log contains information about which user voted which story (item) at which time. If we have user v vote a story about the new iPhone, and shortly later v's friend u does the same, we consider the story as having propagated from v to u, and v as a potential influencer for u. Flixster is one of the main players in the mobile and social movie consumption business. Here, an item is a movie, and the action of the user is rating the movie; an item propagates from v to u, if the v's activation on the item is followed shortly enough by u's activation. Meme is a (discontinued, as of today) microblogging service which allows users to share snippets of text, images, sounds or videos with their social connections. Here the propagations are by means of re-posting information "memes". The *propagation window* Δ is set to 1 month for all the three datasets.

Table 1 and Figure 3 report the main characteristics of the datasets. It is worth noticing how the three datasets exhibits different features. For instance, the first column of Figure 3 reports the distribution of $\mathcal{F}_{i,v}$, that is, for each pair item-user (i, v) the number of possible influencers: we can see that this number is quite low in Digg and high in both Flixster and Meme. Also, the social graph in Flixster is undirected, allowing us to investigate how the CCN technique adapts to undirected graphs.

	Digg	Flixster	Meme
Users	1,000	29,357	9,385
Social Relationships	24,842	425,228	1,144,932
Bidirectional	N	Y	N
Items	31,911	11,659	12,760
Overall Activations($\lvert\mathbb{L}\rvert$)	1,086,065	6,529,011	726,809
Influence Episodes ($\lvert\mathbb{D}\rvert$)	315,377	2,239,744	684,368

Table 1: Summary of the propagation and link data.

Model assessment. There are a number of measurements which can be used to asses the performance of the CCN model. First, the study of the likelihood provides insight on how the models fit the data. Table 2 shows both the penalized log-likelihood of eq. 12 and the *likelihood ratio*. The latter is the ratio between the hypothesized model and an alternative model [13]. In our case, it is the ratio between a given model and the previous model: i.e., 2 communities are compared to 1 community, 4 are compared to 2 and so on. A large value of the ratio denotes that there is a better fitting of the hypothesized model with respect to the previous model. It has been shown that the likelihood ratio approximates the χ^2 distribution for large values of the sample size. Hence, this value gives us an insight of the statistical significance of the model, compared to simpler models.

In a preliminary measurement (not reported here), we found that the ratio between models $K > 1$ and the null model,where $K = 1$, passes the test with 95% significance. This is a clear sign of the presence of community structures in the data. Notably, the test exhibits high positive values on all the cells. The significance of such values can be measured by resorting to the χ^2 test, as mentioned before. What is worth observing here is that both the likelihood ratio and the penalized log-likelihood agree on the optimal number of communities, which is assessed to 8 communities for Digg, and 16 for Flixster and Meme.

[2] `www.isi.edu/~lerman/downloads/digg2009.html`
[3] `http://www.cs.sfu.ca/~sja25/personal/datasets/`

		# Communities				
		2	4	8	16	32
Likelihood Ratio	Digg	5.06E+04	1.50E+05	2.28E+04	2.28E+03	1.82E+03
	Flixster	1.85E+06	2.11E+06	1.65E+06	3.40E+06	4.19E+06
	Meme	1.48E+05	1.80E+05	3.91E+05	1.51E+06	1E+06
Penalized LogLikelihood	Digg	-4.54E+06	-4.41E+06	**-4.40E+06**	-4.45E+06	-4.54E+06
	Flixster	-5.37E+07	-5.21E+07	-5.15E+07	**-5.02E+07**	-5.02E+07
	Meme	-5.08E+07	-5.07E+07	-5.06E+07	**-4.97E+07**	-4.99E+07
Learning Time(Hours)	Digg	1.191	1.87	3.15	6.064	10.81
	Flixster	15.90	20.04	28.73	32.05	59.27
	Meme	15.10	23.89	40.98	74.69	129.73
$\mathcal{Q}_G$/ $\mathcal{Q}_{\mathbb{D}}$	Digg	4.31E-08/2.60E-02	5.11E-08/2.95E-02	6.56E-08/2.98E-02	7.00E-08/3.75E-02	8.16E-08/3.97E-02
	Flixster	7.34E-13/1.72E-03	1.61E-12/1.90E-03	4.35E-12/1.96E-03	7.55E-12/2.00E-03	1.67E-11/2.63E-03
	Meme	5.42E-11/1.22E-03	9.22E-10/1.26E-03	1.30E-09/1.30E-03	1.49E-09/1.36E-03	1.87E-09/1.67E-03

Table 2: Summary of the quality measures.

For the learning time, the table highlights a main issue with the CCN model: a slow (albeit linear) convergence, due mainly to two reasons: (i) the extreme computational burden of the update equations $\delta_{\tilde{u}}^k$ and $\lambda_{\tilde{v}}^k$, and (ii) the fact that the M step is an improvement, rather than an optimization step. The GEM procedure typically exhibits a slower convergence rate than the standard EM procedure: the result is that the learning phase requires more iterations, and each iteration is extremely heavy.

Regarding the first issue mentioned above, it is worth noticing that the main overhead is intrinsic in the exploration of the set of active neighbors who may have potentially triggered the considered activation, in order to determine the most likely influencer. This phase is generally recognized as extremely computing-intensive, and it is a common problem in several method for the estimation of the influence probabilities in a network.

Next, inspired by the well-known *modularity* measure [39], we define two quality measures (for the graph and the propagations log respectively) suitable for our context. In the original formulation, the modularity compares the structure of the graph to that resulting from a random graph, representing a null model. It can be loosely interpreted as follows: for each pair u, v of nodes, the modularity measures whether an actual edge is likely in the hypothesized model. A non-positive weight is associated with unlikely edges, as well as to prospective edges not actually appearing in the graph. Conversely, actual edges which fit the model get a positive weight. We adopt an alternative formulation suitable for our context, but based on a similar principle: actual edges get a positive weight only if they are likely in the model, whereas prospective edges which do not appear in the graph and are likely in the model get a negative weight. The weight is proportional to the probability to observe the edge in the random graph model:

$$\mathcal{Q}_G = \frac{1}{n(n-1)} \sum_{c=1}^{K} P(c) \sum_{u,v \in V} \left[A_{u,v} - P(u,v)\right] r_{uvc} \qquad (13)$$

In the above equation, $A_{u,v}$ is the cell of the adjacency matrix corresponding to the pair (u, v), and $P(u, v) = k_u^{out} k_v^{in}/m^2$ represents the probability of observing the link (u, v) in the null (random graph) model. Also, r_{uvc} is the likelihood of observing the link (u, v) in the community c ($r_{uvc} = \vartheta_u^c \cdot \varphi_v^c$ in the CCN model) and $P(c) = \pi_c$.

A similar formulation can be established for the $\mathcal{Q}_{\mathbb{D}}$ measure relative to the action log $\mathbb{D}$. Following the interpretation of $\mathbb{D}$ as a bipartite graph, the null model is again a random bipartite graph associating items to users. Hence, $P(i, v) = \tilde{k}_i^{out} \cdot \tilde{k}_v^{in}/n_d^2$ where $\tilde{k}_{i_n}^{out}$ is the outdegree of i_n in $\mathbb{D}$, and $\tilde{k}_{v_n}^{in}$ the indegree of v_n. Computing $\mathcal{Q}_{\mathbb{D}}$ for action logs involving a large number of items and users is unpractical, as it requires considering all the possible triplets (i, v, t). A simplified formulation can only consider the actions actually occurring in $\mathbb{D}$:

$$\mathcal{Q}_{\mathbb{D}} = \frac{1}{n_d} \sum_{c} P(c) \sum_{a \in \mathbb{D}} (1 - P(a))\, \tilde{r}_{ac} \qquad (14)$$

where again $\tilde{r}_{ac}$ represents the probability of observing $a = (i, v, t_i(v))$ in c (encoded as $\sum_{u \in \mathcal{F}_{i,v}} \theta_u^{c,a} \cdot \phi_{v,u}^{c,a}$ in the CCN model). Notice that, since we only concentrate on true actions, $\mathcal{Q}_{\mathbb{D}}$ is non-negative and its value increases with the numbers of communities, although we can expect to observe a saturation point.

The values for $\mathcal{Q}_G$ and $\mathcal{Q}_{\mathbb{D}}$ are shown in table 2. We can observe the following:

- the values of $\mathcal{Q}_G$ are always positive. The bias towards 0 is mainly due to the distribution of the values r_{uvc}, which we investigate in the following section.
- The values of $\mathcal{Q}_{\mathbb{D}}$ increase (as expected) with the number of communities, and in general tend to exhibit a higher value than those of $\mathcal{Q}_G$. Again, the distribution of $\tilde{r}_{ac}$ tends to bias the values towards 0.

Analysis. Here we concentrate on the three best models detected in the previous section, and analyse them. We begin with an overall glance at how the models fit the data. To this purpose, for each dataset we plot both the adjacency matrix A and the *influence* matrix I, the latter being the matrix where each pair (u, v) denotes the number of actions $a \in \mathbb{D}$ where u influenced v according to the model, i.e.

$$I_{u,v} = |\{a = (i, v, t) \in \mathbb{D} | u \in \mathcal{F}_{i,v}, u = \operatorname*{arg\,max}_{u' \in \mathcal{F}_{i,v}, 1 \leq k \leq K} \eta_{u,a,k}\}|$$

Rows and columns in both matrices are grouped in buckets, where a user u is associated to a bucket r computed as $r = \arg\max_k P(c_k|u) = \arg\max_k \sum_v \gamma_{u,v,k}$ for the rows and $r = \arg\max_k P(c_k|u) = \arg\max_k \sum_v \gamma_{v,u,k}$ for the columns. The expected results are matrices where both the adjacencies and the influences exhibit block structures with different densities. Indeed blocks are clearly visible in Figure 4. For the Digg dataset, which does nor provide an immediate visual perception, we also produced density plots (Figure 5), where the greyscale in each block represents the density of values within the block. In these latter plots, the structure is once again evident. A closer look at such matrices allows us to get some insights on how the communities are structured. First of all, the matrices reflect a community structure that is inferred by both the action log and the graph structure: indeed, blocks are clearly visible in both the incidence and

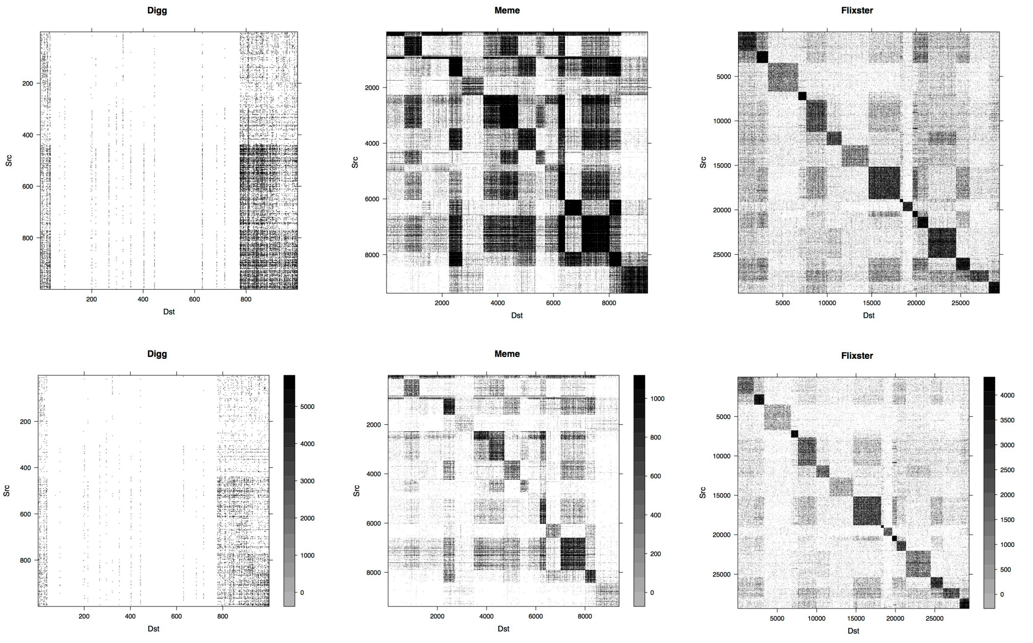

Figure 4: Density plots of the blocks within the adjacency matrix (first row), and of the influence matrix (second row).

the influence matrices. Also, the way blocks are structured is clearly different in the two matrices: for Digg, for example, Figure 5 shows two different blocks of high density: the one relative to the 5th community for the incidence matrix, and the one relative to the 7th community for the influence matrix, respectively. It is also useful to compare the incidence matrix with the one that would result by learning a model without the action log $\mathbb{D}$. Figure 6 shows the blocks that would result in the incidence matrix by running the CCN algorithm without the $\mathbb{D}$ component. A comparison of this matrix with the one in Figure 5 highlights significant differences in the way communities are structured.

For all the three datasets, the matrices exhibit a diagonal structure, a clear indication that users are generally bound to single community. Nevertheless, other blocks can be detected, and they denote the overlapping behavior of some users: since communities model links and actions, some users are likely to assume different roles in more than one community. To this purpose, it is interesting to investigate the distributions of the parameters inferred by the model. We concentrate on Digg only in the following, although similar conclusions can be drawn for the other two datasets.

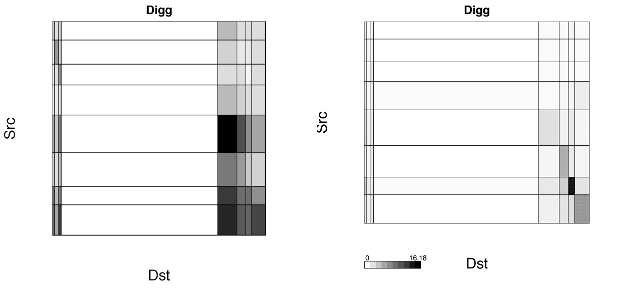

Figure 5: Density plots of the blocks within the adjacency and influence matrices for Digg.

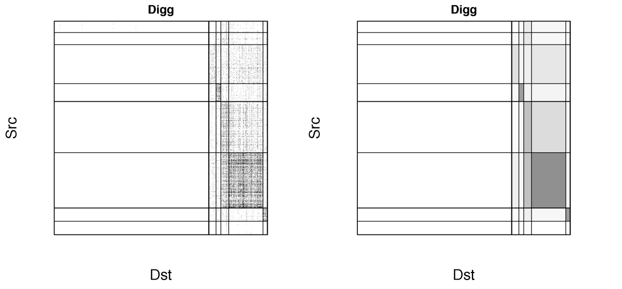

Figure 6: Blocks within the adjacency matrix, resulting by applying CCN to G only: actual values and density regions.

Figure 7 shows such distributions. Notice that, for each community, the value corresponding to a specific user is given by a strip line, where the grey level represent the value in scale. Again, users are sorted according to a given rank, and ranks are assigned by associating each user to the community whose corresponding value is higher (for example, a user u where $\varphi_u^k \geq \varphi_u^{k'}$ is ranked by k in the plot relative to $\vec{\varphi}$).[4] Again, plots exhibit a diagonal structure, where values tend to predominate within a specific community. A similar pattern can be observed also when investigating influence (second row of Figure 7): the first plot shows the frequency distribution actions for influencing users, the second plot (where users are ranked according to $\vec{\vartheta}$) shows how such values distribute for each user and each community. Third and fourth plots report the same information but this time for influenced users.

Finally, Figure 8 plots some characterization of the eight communities found in the Digg dataset. The first plot reports in how many communities each user and each item participate. The histograms show that, while users tend to contribute to few communities, a similar tendency cannot be

[4] The rankings in these matrices are different from the previous ones, because in this case we are interested in detecting peaks in the distributions. In principle, we expect that each community is characterized by a kernel of users whose values dominate over those of the others.

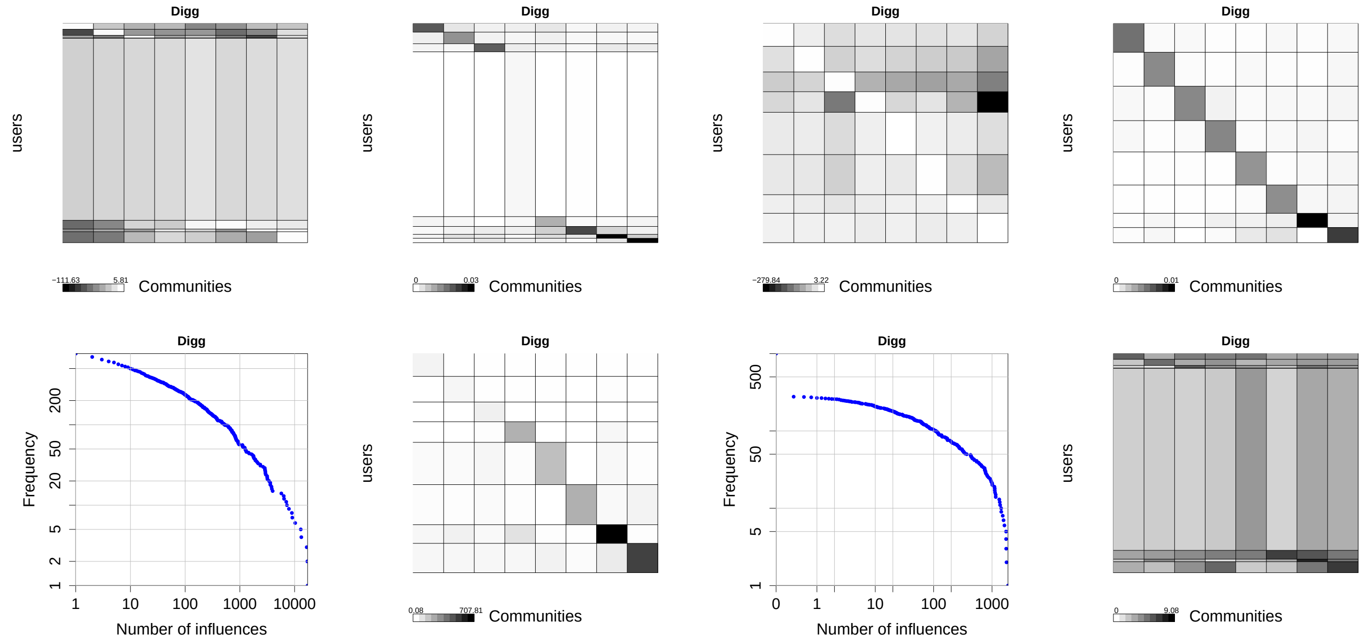

Figure 7: First Row: Distribution of the model parameters: Π^d, $\vec{\varphi}$, Π^s and $\vec{\vartheta}$. Second row: Frequency distributions for influence: influencing (first two plots) and influenced users (remaining plots).

detected for items involved in actions, which spread across different communities. The second plot shows how nodes, links, and items populate each of the eight communities. We can observe that the communities found are quite equilibrated in this respect. Note that in this histogram nodes and items can simultaneously appear in multiple communities, whereas links are associated to a single community.

5. CONCLUSIONS AND FUTURE WORK

The CCN model proposed in this paper provides a simple probabilistic framework for jointly modeling the community structure of a social network and the information cascades happening on the same social network. The key intuition is that each observed action (e.g, the creation of a follower-followed link in `Twitter`, or the re-tweeting of a meme) can be explained by the same hidden factors, namely the level of "active involvement" $\pi_u^{c,s}$ of user u in a topic/community c, and the degree of "passive involvement" $\pi_u^{c,d}$. The experimental analysis on three real-life datasets confirms our intuition.

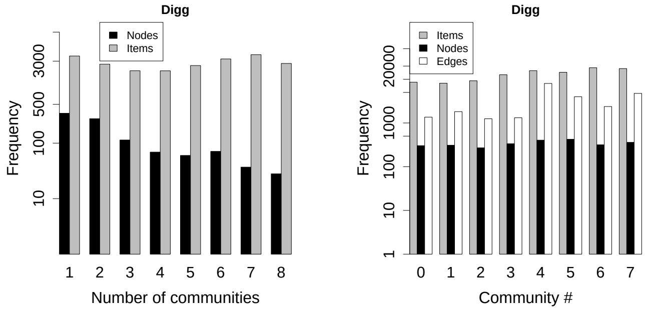

Figure 8: Distribution of the number of communities per item/node (left), nodes, links and items population for each of the eight communities (right).

We believe that, thanks to its simplicity, our CCN model has a wide range of potentialities still unexplored. By modeling communities and cascades formation jointly, CCN might be a useful tool in the analysis of both social phenomena, potentially providing interesting insights that cannot be obtained by analysing the two phenomena in isolation.

Being a stochastic generative model, CCN lends itself to a variety of predictive tasks. As a matter of fact, there are several application scenarios where it is important to predict whether a user shall adopt a given item, or whether two users are likely to be connected. For instance, predicting whether a user will be influenced and will participate in a cascade is a key step in *influence maximization* [28] for viral marketing. Predicting links formation instead can be used for user-to-user recommendation, as well as to the analysis of the evolution of the social networks. Since CCN models directly such probabilities, it is important to evaluate the accuracy of such predictions. In our previous studies [8, 9], we already showed that topic modeling improves the predictive accuracy for item adoption. We expect similar findings for CCN, and we plan to assess this in our future work.

A main drawback of the CCN model is the slowness of the learning phase. However, some of the causes of such slowness are structural and related to the general problem of estimating influence strength. Under this perspective, we plan to study how to parallelize the learning phase.

Finally, it is important to devise a extension of the CCN approach which relies on a bayesian treatment: since large graphs and action logs are sparse data, a treatment of priors can make the technique more robust, even in the presence of extreme values.

Acknowledgments. This research was partially supported by the Torres Quevedo Program of the Spanish Ministry of Science and Innovation, and partially funded by the European Union 7th Framework Programme under grant n. 270239 (ARCOMEM).

6. REFERENCES

[1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD*, 2008.

[4] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proc. of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[5] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *WSDM*, 2011.

[6] B. Ball, B. Karrer, and M. E. J. Newman. An efficient and principled method for detecting communities in networks. *Phys. Rev. E 84, 036103*, 2011.

[7] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. Magdon-Ismail, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. In *IADIS AC*, 2005.

[8] N. Barbieri, and G. Manco. An Analysis of Probabilistic Methods for Top-N Recommendation in Collaborative Filtering. In *ECML PKDD*, 2011.

[9] N. Barbieri, G. Manco, R. Ortale and E. Ritacco alancing Prediction and Recommendation Accuracy: Hierarchical Latent Factors for Preference Data. In *SDM*, 2012.

[10] F. Bonchi. Influence propagation in social networks: A data mining perspective. *IEEE Intelligent Informatics Bulletin, Vol.12 No.1: 8-16, December 2011.*

[11] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen. Chromatic Correlation Clustering. In *KDD*, 2012.

[12] F. A. Breve, L. Zhao, M. G. Quiles, W. Pedrycz, and J. Liu. Particle competition and cooperation for uncovering network overlap community structure. In *ISNN*, 2011.

[13] G. Casella and R. Berger. *Statistical Inference*. Duxbury Press, 2001.

[14] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2010.

[15] W. Chen, Z. Liu, X. Sun, and Y. Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Min. Knowl. Discov.*, 21(2):224–240, 2010.

[16] D. J. Crandall, D. Cosley, D. P. Huttenlocher, J. M. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, 2008.

[17] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, 2001.

[18] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[19] T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Phys. Rev. E*, 80:016105, 2009.

[20] T. L. Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW*, 2010.

[21] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2010.

[22] R. Ge, M. Ester, B. J. Gao, Z. Hu, B. K. Bhattacharya, and B. Ben-Moshe. Joint cluster analysis of attribute data and relationship data: The connected *k*-center problem, algorithms and applications. *TKDD*, 2(2), 2008.

[23] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[24] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.

[25] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.

[26] R. Guimerà, M. Sales-Pardo, and L. Amaral. Module identification in bipartite and directed networks. *Phys. Rev. E*, 2007.

[27] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2):256–276, 2006.

[28] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[29] Y. Kim and H. Jeong. The map equation for link communities. *Phys. Rev. E 84, 026110*, 2011.

[30] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Phys. Rev. E*, 78:026109, 2008.

[31] A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 11:033015, 2009.

[32] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 04 2011.

[33] S. Lattanzi and D. Sivakumar. Affiliation networks. In *STOC*, 2009.

[34] E. Leicht and M. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100(11), 2008.

[35] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *TWEB*, 1(1), 2007.

[36] J. Leskovec, A. Singh, and J. M. Kleinberg. Patterns of influence in a recommendation network. In *PAKDD*, 2006.

[37] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, 2009.

[38] R. M. Neal and G. E. Hinton. *A view of the EM algorithm that justifies incremental, sparse, and other variants*, pages 355–368. MIT Press, Cambridge, MA, USA, 1999.

[39] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[40] A. Padrol-Sureda, G. Perarnau-Llobet, J. Pfeifle, and V. Muntés-Mulero. Overlapping community search for social networks. In *ICDE*, 2010.

[41] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043), 2005.

[42] G. Palla, I. Farkas, P. Pollner, I. Derenyi, and T. Vicsek. Directed network modules. *New J. Phys.*, 9(6), 2007.

[43] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, 2007.

[44] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.

[45] D. M. Romero, B. Meeder, and J. M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, 2011.

[46] M. Sachan,D. Contractor,T Faruquie, and L.V. Subramaniam Using content and interactions for discovering communities in social networks. *WWW*:331–340, 2012.

[47] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES*, 2008.

[48] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[49] A. Silva, W. M. Jr., and M. J. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB*, 5(5):466–477, 2012.

[50] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, 2009.

[51] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *KDD*, 2010.

[52] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.

[53] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, 2010.

[54] J. Xie, S. Kelley, and B. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Survey*, 45(4), 2013.

[55] J. Xie and B. K. Szymanski. Towards linear time overlapping community detection in social networks. In *PAKDD*, 2012.

[56] J. Xie, B. K. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *ICDM Workshops*, 2011.

[57] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.