Detecting Communities Using Information Flow in Social Networks

David Darmon[1], Elisa Omodei[2], Cesar Flores[3], Luís Seoane[4], Kevin Stadler[5], Jody Wright[6], Joshua Garland[7], and Nix Barnett[8]

[1]Department of Mathematics, University of Maryland, College Park

16 September 2013

Abstract

Many complex networks are characterized by a community structure, i.e. the presence of groups of nodes that are more densely connected among each other than with the rest of the network. The algorithms developed to detect such communities usually consider the structural properties of the network, i.e. the static links between nodes. In the case of social networks this means considering for example "friendship" links on Facebook or "followers" on Twitter. We argue that these kind of links are not indicative of the real community structure underlying these networks, since social network users usually have hundreds of connections even with people they are only acquaintances with, but they communicate and exchange only with a subset of these connections, and form real communities only within these subsets. In this paper, we adapt standard community detection algorithms to account for this reality using recent work in information theoretic network analysis. We apply this approach to detect *dynamical / functional* communities in both synthetic and empirical networks, and determine how the composition of the communities change over time. We find that by explicitly incorporating the observed dynamics of users in social media, we can identify communities hidden in the structural network.

1 Introduction

Lorem ipsum.

2 Methods

2.1 Network and Dynamics

Put a blurb here about the distinction between 'dynamics on vs dynamics of' with networks?

First, we consider the most general case of a dynamics *on* a *structurally-static* network. We fix some notation, following [7]. Let $G = (V, E)$ be the graph that represents the $|V|$ vertices in the network and the $|E|$ edges between them. We will consider a random field evolving in time on this network. That is, for a finite network, let $X(t, v)$ denote the random variable in some countable alphabet $\mathcal{X}_v$ associated with vertex v at time t. Overall, $X(t, v)$ is a time-varying random field whose dynamics take place on G and are effected by the topology of G. Thus, for fixed t, $X(t, \cdot)$ is a random vector and for fixed v, $X(\cdot, v)$ is a random field. We will occasionally refer to the random field at a fixed timepoint t as $\mathbf{X}(t) = (X(t, v_1), \ldots, X(t, v_{|V|}))$.

For the real world problem we consider in this paper, G corresponds to an (explicit/structural) social network, and $X(t, v)$ corresponds to some observed behavior of an individual v at time t. Since the empirical data consists of the tweeting behavior of users on Twitter, we will frequently fix $\mathcal{X}_v = \{0, 1\}$ with $X(t, v) = 1$ indicating that user v tweeted at time instant t, and $X(t, v) = 0$ indicating that user v did not tweet at time instant t.

2.2 Community Structure

Many real networks have a natural community structure, where disjoint subgroups of nodes exchange more connections within their subgroup than between subgroups. Formally, we want to compute the optimal division of the network that minimizes the number of links between subgroups (also called communities). The raw number of links across boundaries of communities does not give a good partition of the network. For example, the community structure can be a consequence of random variations in the density of links. A more reliable approach uses the configuration model [11] as a null model to assess the quality of a given network partition. Newman and Girvan [10] define the modularity as follows:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(g_i, g_j) \tag{1}$$

where $m = \frac{1}{2} \sum_{ij} A_{ij}$ is the number of links and g_i indicates the label of the community the node i belongs to. Notice that maximizing the above function yields a partition that minimizes the expected number of links falling between different communities, i.e., when $\delta(g_i, g_j) = 0$. Modularity Q takes values between 0 and 1: low modularity indicates the number of links between distinct communities is not significantly different from the random distribution and high modularity indicates there is a strong community structure.

2.3 Edge Weighting using Dynamics

Put a brief rationale for why we use this approach. Or perhaps this goes in an introductory section where we argue for the need to determine com-

munities based on dynamics?

Let $X^u = X(\cdot, u)$. That is, in what follows we implicitly assume the stationarity of $X(t, v)$ with respect to time[1]. Denoting the joint distribution of (X^u, X^v) as $p(x^u, x^v)$ and the associated marginals as $p(x^u)$ and $p(x^v)$, we then define the mutual information between two individuals in the usual way [2] as

$$I[X^u; X^v] = E\left[\log_2 \frac{p(X^u, X^v)}{p(X^u)p(X^v)}\right] \tag{2}$$

$$= \sum_{x^u \in \mathcal{X}^u, x^v \in \mathcal{X}^v} p(x^u, x^v) \log_2 \frac{p(x^u, x^v)}{p(x^u)p(x^v)}. \tag{3}$$

The mutual information is not generally bounded on a standard interval. To allow for standardized weightings, we follow [14] and normalize the mutual information, noting that

$$I[X^u; X^v] = H[X^u] - H[X^u|X^v] \leq H[X^u] \tag{4}$$

(by the non-negativity of $H[X^u|X^v]$), and equivalently

$$I[X^u; X^v] = H[X^u] - H[X^u|X^v] \leq H[X^v]. \tag{5}$$

Overall, this implies that

$$I[X^u; X^v] \leq \min\left\{H[X^u], H[X^v]\right\}, \tag{6}$$

so dividing by $\min\left\{H[X^u], H[X^v]\right\}$ gives the normalized mutual information

$$I^*[X^u; X^v] = \frac{I[X^u; X^v]}{\min\{H[X^u], H[X^v]\}} \tag{7}$$

which lies between 0 and 1.

Of course, in an empirical study, we do not know the information theoretic quantities associated with any two users. Instead, we must infer them from an observed time series. To do so, we use the maximum likelihood estimates for all quantities [12]. In the absence a parametric model (which we do not assume here), this amounts to using the plug-in estimator for $p(x^u, x^v)$ in (7). Assuming we observe $\{(X(t, u), X(t, v))\}_{t=1}^T$, the plug-in estimator for $p(x^u, x^v)$ is simply

$$\hat{p}(x^u, x^v) = \frac{\#(X(t, u) = x^u, X(t, v) = x^v)}{T}, \tag{8}$$

the proportion of times we observe a particular behavior in both individuals out of all behaviors observed.

As observed in [12], all estimators for information theoretic quantities are biased, including the plug-in estimator. However, the bias is generally proportional to the size of the joint alphabet $\mathcal{X}^u \times \mathcal{X}^v$ and inversely proportional to the sample size T. Since we will typically take $\mathcal{X}^u = \{0, 1\}$ for all individuals u, $|\mathcal{X}^u \times \mathcal{X}^v| = 4$, and the bias will be small for the sample sizes we consider.

How this gets used in a community detection algorithms goes here (or in a nearby section).

[1] If we do not assume stationarity, the statistics we compute have meaning but are no longer estimators for the parameters we describe. See [16].

2.4 Community detection across time snapshots

In addition to study the community structure a static single network, we studied how the community structure changes across time snapshots. The main reason of this analysis is that, in real Twitter data, the mutual information between users may vary across time, and therefore the community structure of the mutual information network may change across time. As a specific example, this type of analysis could potentially be a method to detect the birth of trending topics in the Twitter social network.

In order to perform the analysis we used two modularity algorithms specifically designed either for time snapshots or multiplex networks (a time varying network can be described as a multiplex network in which each time snapshot is a multiplex layer). These two algorithms are described on the articles published by Kawadia and Sreenivasan [6], and Mucha et Al [9]. Both algorithms are available online as open source in Python and Matlab, respectively.

2.4.1 Kawadia and Screenivasan algorithm

The idea of this algorithm is to find a new community partition P_t for a graph snapshot G_t, such that the estrangement (a distance metric that the authors defined) between the new partition and the previous one is smaller than δ. In other words, the algorithm solves the following constrained optimization problem:

$$\begin{array}{l} \text{maximize}_P \quad Q(P) \\ \text{subject to } E(P) \leq \delta, \end{array} \tag{9}$$

where $E(P)$ represents the estrangement between the current and the last community partition. The authors make use a Lagrange multiplier to solve this problem (for more details see [6]). Unfortunately, even that the algorithm worked for the synthetic data, it did not for the real Twitter data. The reason is that the Python library that they use to solve the constraint satisfaction problem becomes really slow for networks of big size as is the case of the real Twitter data that we present here.

2.4.2 Mucha et al. algorithm

This algorithm is based in optimizing a modularity definition that includes extra terms for the existence of different network layers, which in this case consist of different time snapshots. The new modularity definition that the authors propose to optimize in order to find the change of modularity across layers is:

$$Q_{\text{multilayer}} = \frac{1}{2\mu} \sum_{ijsr} \left\{ \underbrace{\left(A_{ijs} - \gamma_s \frac{k_{is} k_{js}}{2m_s} \right) \delta_{sr}}_{\text{within layer contribution}} + \underbrace{\delta_{ij} C_{jsr}}_{\text{between layer contr.}} \right\} \delta(g_{is}, g_{jr}), \tag{10}$$

where A_{ijs} is the element i, j of the adjacency matrix of layer s, k_{is} the degree of node i inside layer s, m_s the total weight of layer s, g_{is} the module index of node i inside layer s. Finally, the user parameter γ_s balances the within layer contribution and C_{jsr} balances the between layer contribution. C_{jsr} can be seen as a virtual link between node j's of two different layers s and r.

In order to optimize the modularity equation, the authors use a generalization of the Louvain algorithm [1], which is one of the most used when dealing with large scale networks. The released Matlab library of this algorithm outperformed the previous one in terms of performance and speed. Therefore, this algorithm is the one that we used to find how modularity changes across time.

3 Examples

We apply the methodology developed above to two cases: a toy model of user dynamics and a real world dataset from Twitter.

3.1 Coupled Bernoulli Processes Embedded in a Structural Network Drawn from a Stochastic Block Model

Describe the toy model in generalities.

3.1.1 Stochastic Block Model

We use the stochastic block model [5] as a generative model for the structural network. The basic stochastic block model is a popular model for simple community structure. We will specify the community of a user u by $C(u)$ and the set of all communities by $\mathcal{C}$. Each individual u in the network is assigned to a (latent) community $C(u) = c \in \mathcal{C}$. Edges are then placed between each user u and v with probability $p_{uv} = p_{C(u),C(v)}$ depending only on the membership the two users. Typically, for assortative communities, $p_{cc} > p_{cc'}$ for all $c \neq c' \in \mathcal{C}$. That is, the density of links within a community is greater than the density of links between communities, which is the standard definition of community structure. These edge probabilities can be collected into a matrix $\mathbf{P}$ where $(\mathbf{P})_{cc'} = p_{cc'}$. If we fix $p_{cc} = p_{\text{in}}$ and $p_{cc'} = p_{\text{out}}$ for all $c \neq c' \in \mathcal{C}$, we obtain an edge probability matrix like Figure 1. This clearly posses a 'block' structure (hence the name stochastic block model) with clear communities. The adjacency matrix for a particular realization from this model resembles Figure 2.

3.1.2 Coupled Bernoulli Process

Previous work [15] has considered modeling users on Twitter as coupled Poisson processes. In this model, the users are embedded in a directed network, with each vertex in the graph corresponding to a user and each directed edge indicating the presence of 'influence' of the initiating user on the terminal user. Influence was modeled as follows: after a user u tweets, the user exerts an influence on its directed neighbors by increasing the instantaneous rate of their associated Poisson process for some interval of time. The authors of [15] took the coupling term to decay as a reciprocal power of the time since the tweet occurred.

In this work, we consider a modified version of this model. First, since communication on digital social networks such as Twitter occur in discrete time, we explicitly model each user as a *Bernoulli* process, the discrete-time analog of a Poisson process. Second, in this initial work, we only consider an influence to occur over a single time delay. This corresponds to a choice of time scale. Thus, at a given time instant t, the

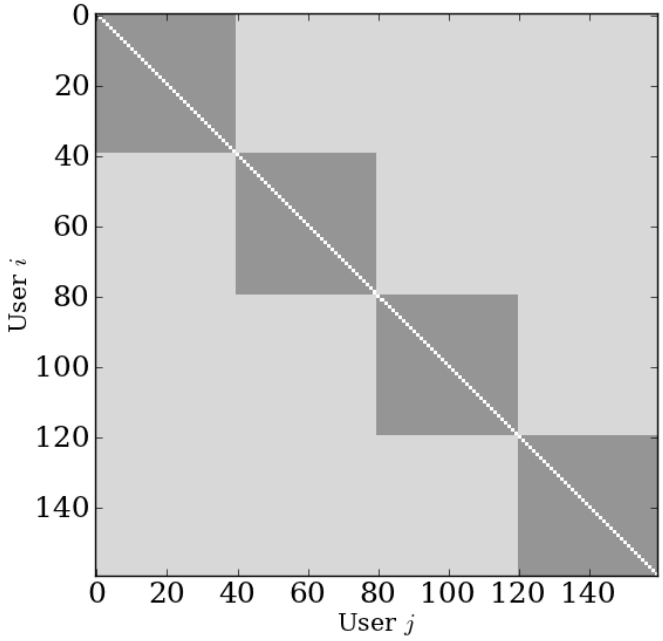

Figure 1: The edge probability matrix $\mathbf{P}$ for a stochastic block model with four communities each with 40 members.

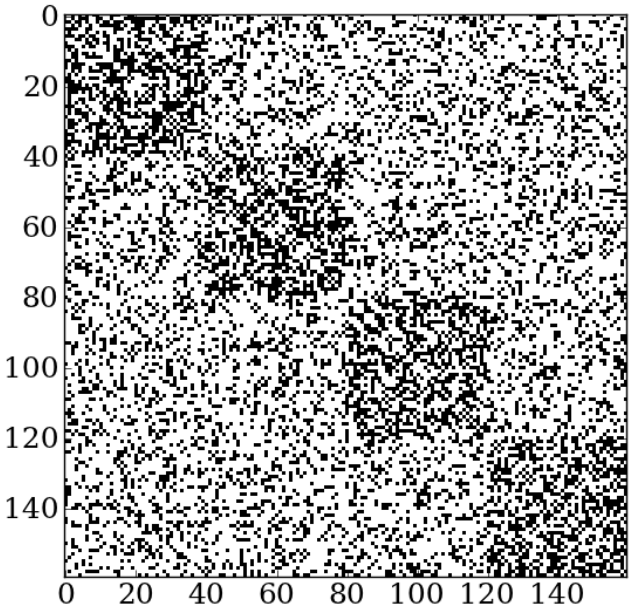

Figure 2: The (undirected) adjacency matrix $\mathbf{A}$ from a realization of the stochastic block model specified by the edge probability matrix $\mathbf{P}$ from Figure 1.

probability of a particular user tweeting, given the entire past behavior of all of the users is

$$P(X(t,v)=1|\mathbf{X}(t-1),\mathbf{X}(t-2),\ldots) = P(X(t,v)=1|\{X(t-1,u):u\in\mathcal{N}(v)\}) \tag{11}$$

$$= \min\left\{p_v + \sum_{u\in\mathcal{N}(v)} \iota_{uv} 1[X(t-1,u)=1], 1\right\} \tag{12}$$

where $\mathcal{N}(v)$ denotes the directed neighbors of v (those users in the network with directed edges from themselves to v), and ι_{uv} denotes the influence of user u on user v.

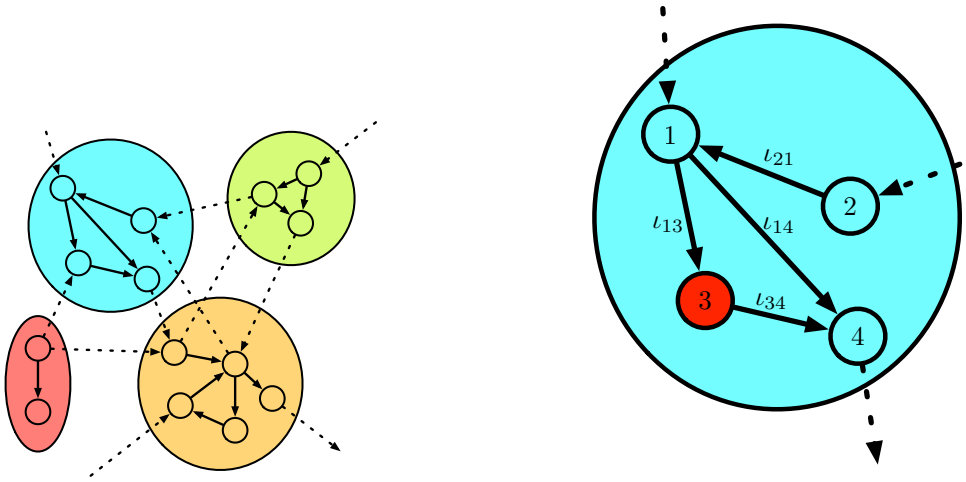

Figure 3: Schematics for the coupled Bernoulli model. Left: A collection of communities defined by the dynamics of their users. Right: Focusing on the influencers of a particular user.

A schematic for this model is shown in Figure 3. In the left schematic, each solid circle corresponds to a set of nodes in a particular dynamical / functional community. Note that these communities may differ from the *structural communities* as defined by the stochastic block model in Section 3.1.1. In particular, we will take ι_{uv} to be larger between two nodes u and v within the dynamical community than between two nodes in different dynamical communities. In the right schematic, we focus on user 3. For this user, we see that the equation determining the probability of that user tweeting at time t (Equation (12)) reduces to

$$P(X(t,3)=1|\mathbf{X}(t-1),\ldots) = P(X(t,3)=1|X(t,1)) \tag{13}$$

$$= \min\{p_3 + \iota_{13}1[X(t-1,1)=1], 1\} \tag{14}$$

$$= \text{base rate } + \text{ influence}, \tag{15}$$

a base probability plus the influence of 3's directed neighbors.

3.1.3 Choice of Parameters for Toy Model

For the results presented in this paper, we took $p_{\text{in}} = 0.1$ and $p_{\text{out}} = 0.05$. These values were chosen to be below the detectability threshold [3, 8]. For values of p_{out} close to p_{in} (assuming that $p_{\text{in}} > p_{\text{out}}$, as would be the case for assortative social networks), community structure cannot be reliably inferred. Thus, while structural communities are present, any structural communities inferred using the structural network alone will be spurious. For each edge generated by the stochastic block model, the direction of the edge was chosen at random. We took $\iota_{\text{in}} = \frac{0.7}{4}$ and $\iota_{\text{out}} = \frac{0.07}{4}$. Thus, the influence of users within a dynamical community is ten times the influence of users outside the dynamical community. The value for ι_{in} was chosen to be below a critical value ι_{in}^* (dependent on p) that lead the users in each community to constantly tweet.

3.2 Twitter Dataset

The data consists of the Twitter statuses of 12,043 users over a 49 day period. The users are embedded in a 15,000 node network collected by performing a breadth-first expansion from a seed user. Once the seed user was chosen, the network was expanded to include his/her followers, only including users considered to be active (users who tweeted at least once per day over the past one hundred tweets). Network collection continued in this fashion by considering the active followers of the active followers of the seed, etc.

The statuses of each user were transformed into a binary time series using their time stamp as follows. For each user u, we consider only the relative times of their tweets with respect to a reference time. Denote these times by $\{\tau_j^u\}_{j=1}^{n_u}$. Let the reference start time be t_0 and the coarsening amount be Δt. From the tweet times, we can generate a binary time series $\{X(i,u)\}_{i=1}^{T}$, where

$$X(i,u) = \begin{cases} 1 & : \exists \tau_j^u \in [t_0 + (i-1)\Delta t, t_0 + i\Delta t) \\ 0 & : \text{ otherwise} \end{cases} . \tag{16}$$

In words, $X(i,u)$ is 1 if user u tweeted at least once in the time interval $[t_0 + (i-1)\Delta t, t_0 + i\Delta t)$, and 0 otherwise. Because the recorded time of tweets is restricted to a 1-second resolution, a natural choice for Δt is 1 second. However, computing mutual information between second-resolution tweet series would neglect medium- to long-range influences between individuals. Thus, we generally take Δt to be on the order of minutes.

In this paper, only tweets made between 7 AM and 10 PM (EST) were considered. For any second during this time window, a user either tweets, or does not. Thus, each day can be considered as a binary time series of length 57,600, with a 1 at a timepoint if the user tweets, and a 0 otherwise.

4 Results

4.1 Community Detection — Structural vs. Dynamical Links

In order to test our hypothesis we ran a community detection algorithm on the two networks under analysis: the one produced by the toy model and the Twitter one. We

used the algorithm developed by Blondel at al. [1], which is found to be one of the best algorithms for community detection based on modularity optimization, as shown by Fortunato in his review of communities detection methods[4]. The algorithm works in two phases: in the first one the modularity is optimized by allowing only local changes of communities, and in the second one the found communities are aggregated in order to build a new network of communities. These two phases are repeated iteratively until no increase of modularity is possible. In the future we plan to expand the analysis by using also other algorithms, such as Infomap[13], and compare the obtained results. The analysis was carried out in two steps: first we applied the algorithm to the binary unweighted network, considering in this way only the *structural* relationships between the nodes. These are given, in the case of the real data, by looking at who is *following* who on Twitter, which is an explicitly declared relationship. The limitation of this method is that most Twitter users follow a very large number of people, but in practice they see and share only the tweets belonging to a relatively small subset of the users they follow. Therefore the second step was to apply the algorithm to a weighted version of the networks, that we obtained by weighting each structural link with the value of the mutual information between the two users. The results of this analysis are shown in Table 1.

Table 1: Communities Detection Results

	Synthetic Network		Twitter Network	
	Binary	Weighted	Binary	Weighted
Number of Detected Communities	8	4	26	37
Optimal Modularity Value	0.264	0.625	0.260	0.404

We can observe a clear increase in the optimal value of modularity when we consider the weighted network, which indicates that the network has a more defined community structure. In the case of synthetic data the value triples, whereas in the case of real data it doubles. This result shows, as we expected, that if we weight the structural links of a network with a measure of the communication flow among the nodes, giving thus more importance to the links between two users that actually share information and less to the links that are there but are not exploited, we can detect a finer community structure. In particular, for what concerns the synthetic data, we can observe that in the weighted case we find exactly four communities, which is coherent to the way the toy model was built. A visual representation of the results is given in Figure 4 for the synthetic data and in Figure 5 for the Twitter data.

Because the representation of the network is different (binary vs weighted), the modularity value is not enough to say that the community structure is different. The differing values could be due to computing the modularity with and without weighting. To test this hypothesis, we consider the actual community structure in both cases. The sizes and overlap of the largest communities are shown in Table 2. By just looking at the number of modules in each representation, we can be tempted to say that the community structure is different. However, as the next table shows, the percentage of total nodes inside the biggest five nodes is very high. In other words, the first five biggest modules dominates the community structure.

Table 2: Lorem ipsum.

	1	2	3	4	5	Total %
Binary	1092	941	410	292	12	97.76
Weighted	1098	938	220	293	140	95.69
Shared	876	777	66	244	0	69.86

First, notice that the total size (last column) of the five biggest modules very high (> 95 %) as already described. Second, notice that for the binary row we sorted the modules in decreasing order. However, for the weighted row this is not the case (see module 3 and 4). We relabel the weighted partition in order to maximize similarity structure with the binary partition.

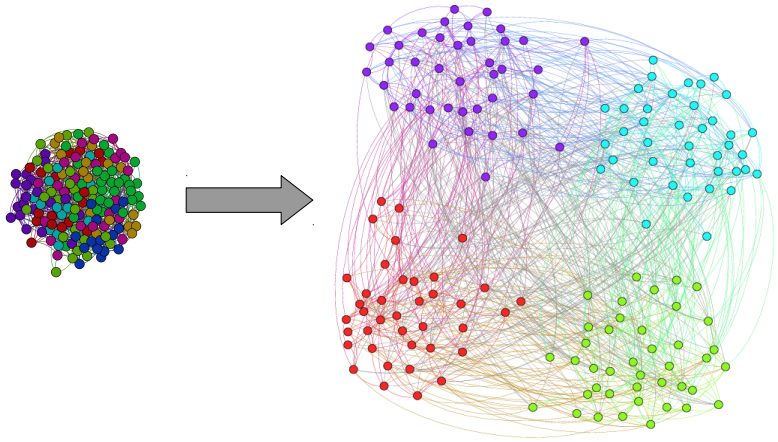

Figure 4: Visualisation of the synthetic network obtained using binary edges (left figure) and weighted ones (right figure). Node colors indicate the community affiliation.

4.2 Community Detection Across Time

Figure 6 represent the modularity across the seven different weeks for a single combination of parameters. This Figure shows, that in essence, the structure does not vary across time. This fact tell us that for this data it may be better to just evaluate the modularity of a single snapshot across the entire seven weeks.

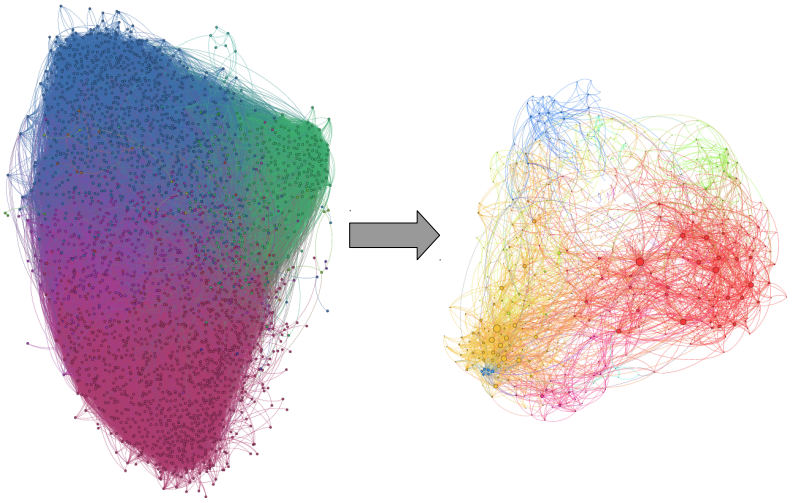

Figure 5: Visualisation of the Twitter network obtained using binary edges (left figure) and weighted ones (right figure). Node colors indicate the community affiliation.

5 Discussion

Lorem ipsum.

6 Conclusions and Future Work

Lorem ipsum.

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[2] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[3] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701, 2011.

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

[5] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

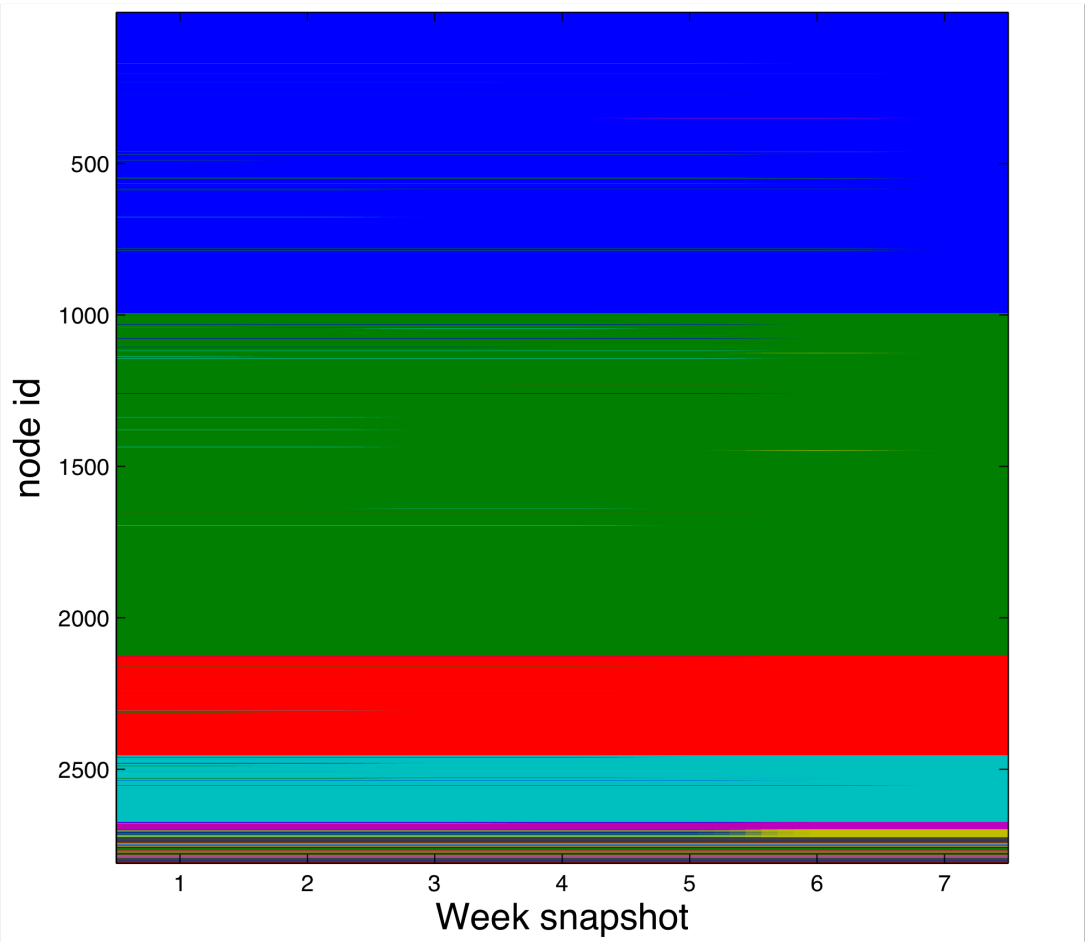

Figure 6: Modularity across time with parameters $\gamma_s = 1$ for all s and $C_{jrs} = \omega = 0.25$ for all j, s, r. Different colors represent the module id to which each node (rows in this figure) belong to. We can observe, that the structure is dominated by the existence of four big modules plus a lot of small size modules (bottom of the Figure). Further, the modularity structure remains almost constant across time in the sense that only a very small number of nodes (represented as colored lines inside the big modules) switch to other modules as time develops.

[6] Vikas Kawadia and Sameet Sreenivasan. Sequential detection of temporal communities by estrangement confinement. *Scientific reports*, 2, 2012.

[7] Eric D Kolaczyk. *Statistical analysis of network data: methods and models*. Springer, 2009.

[8] Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499*, 2012.

[9] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[10] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[11] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[12] Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

[13] Martin Rosvall and Carl T Bergstrom. Mapping change in large networks. *PloS one*, 5(1):e8694, 2010.

[14] Cosma Rohilla Shalizi, Marcelo F Camperi, and Kristina Lisa Klinkner. Discovering functional communities in dynamical networks. In *Statistical network analysis: Models, issues, and new directions*, pages 140–157. Springer, 2007.

[15] Greg Ver Steeg and Aram Galstyan. Information transfer in social media. In *Proc. 21st Int'l World Wide Web Conf.*, pages 509–518. ACM, 2012.

[16] Vincent Q Vu, Bin Yu, and Robert E Kass. Information in the nonstationary case. *Neural computation*, 21(3):688–703, 2009.