Detecting Cohesive and 2-mode Communities in Directed and Undirected Networks

Jaewon Yang, Julian McAuley, Jure Leskovec
Stanford University
{jayang, jmcauley, jure}@cs.stanford.edu

ABSTRACT

Networks are a general language for representing relational information among objects. An effective way to model, reason about, and summarize networks, is to discover sets of nodes with common connectivity patterns. Such sets are commonly referred to as *network communities*. Research on network community detection has predominantly focused on identifying communities of densely connected nodes in undirected networks.

In this paper we develop a novel overlapping community detection method that scales to networks of millions of nodes and edges and advances research along two dimensions: the connectivity structure of communities, and the use of edge directedness for community detection. First, we extend traditional definitions of network communities by building on the observation that nodes can be densely interlinked in two different ways: In *cohesive* communities nodes link to each other, while in *2-mode* communities nodes link in a bipartite fashion, where links predominate *between* the two partitions rather than inside them. Our method successfully detects both 2-mode as well as cohesive communities, that may also overlap or be hierarchically nested. Second, while most existing community detection methods treat directed edges as though they were undirected, our method accounts for edge directions and is able to identify novel and meaningful community structures in both directed and undirected networks, using data from social, biological, and ecological domains.

Categories and Subject Descriptors: H.2.8 **[Database Management]**: Database Applications – *Data mining*
General Terms: Algorithms, theory, experimentation.
Keywords: Network communities, Overlapping community detection, 2-mode communities.

1. INTRODUCTION

Networks are a powerful way to model relational information among objects from social, natural, and technological domains. Networks can be studied at various levels of resolution ranging from whole networks to individual nodes. Arguably the most useful level of resolution is at the level of groups of nodes. Studying groups of nodes allows us to identify and analyze modules or components of networks. For example, understanding the organization of networks at the level of groups helps us to discover functional roles of proteins in protein-protein interaction networks [36], political factions in a network of bloggers [1], social circles in online social networks [31], or even topics in word association networks [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WSDM'14, February 24–28, 2014, New York, New York, USA.
Copyright 2014 ACM 978-1-4503-2351-2/14/02 ...$15.00.
http://dx.doi.org/10.1145/2556195.2556243.

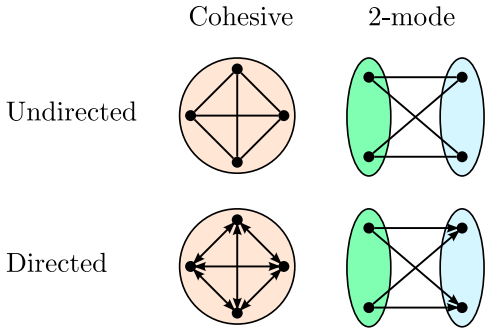


Figure 1: Two types of networks (directed and undirected) and two types of communities (cohesive and 2-mode). While research has predominantly focused on undirected-cohesive communities (top left), we develop a method that can detect cohesive as well as 2-mode communities in both directed and undirected networks.

One way to understand networks at the level of groups is to identify sets of nodes with similar connectivity patterns. Traditional methods aim to find network *communities*, which are defined as groups of nodes with many connections among the group's members, but few to the rest of the network [2, 11, 14, 34]. However, dense communities are but one kind of group structure in networks, and there may be other structures that help us to understand networks better. For example, consider a Twitter follower network and the "community" of candidates in the 2012 U.S. presidential election. This community is not densely interlinked, in the sense that the candidates do not follow each other; thus we would not be able to find this community if we were to use traditional methods that search for densely connected sets of nodes. However, such communities can be identified because they form around nodes whose edges have similar endpoints. Continuing our example, presidential candidates form a community in Twitter not because they follow each other but because a common set of "fans" follows them.

Thus communities can be characterized by the connectivity structure between the members *and also* by the connectivity structure of the members to the rest of the network. We refer to these communities as *2-mode communities*. For example, in case of "fans" linking to "celebrities" members of a community may be linked to the same set of endpoints, even if they do not link to each other. Similar examples also exist beyond social networks; for example, in protein-protein interaction networks, some protein complexes act

as bridges or regulators, *i.e.*, they do not interact among themselves but regulate/interact with the same set of proteins [36].

Another common assumption made by many present community detection methods is that networks are *undirected* [34, 43]. This implies that relationships between connected nodes are symmetric or reciprocal. However, in directed networks relationships are asymmetric, as with our previous example about "fans" who follow "celebrities". Even though methods can often be adapted to handle directed networks, this is often done in an ad-hoc fashion (*e.g.*, by treating directed edges as though they were undirected) and can lead to unexpected or undesirable results [14, 26, 39]. Moreover, by ignoring edge directedness important information may be lost, especially if relationships are predominantly non-reciprocal as in predator-prey networks [26] or in social networks like Twitter.

Present work: Detecting cohesive and 2-mode communities in directed and undirected networks. Here we consider new notions of community linking structure that go beyond thinking of communities as internally well-connected sets of nodes. Our work stems from social network literature on structural equivalence [9], where it has been noted that social homogeneity (*i.e.*, social communities) arises not only between nodes that link to each other (*i.e.*, *internal group connectivity*), but also between nodes that link to the rest of the network in a coordinated way (*i.e.*, *external group connectivity*). In particular, we consider different notions of "communities" that are depicted in Figure 1. We differentiate between *cohesive* communities (Fig. 1, Cohesive) and *2-mode* communities (Fig. 1, 2-mode) where nodes link in a bipartite fashion with links predominantly appearing between partitions rather than inside them.

While existing community detection methods typically focus on Undirected-Cohesive or Directed-Cohesive communities [11, 14, 26, 34, 43], the focus of our paper is on developing methods that can detect communities of all four different types depicted in Figure 1. By modeling each of these definitions in concert, we are able to capture the complex structure present in networks.

Present work: Communities through Directed Affiliations. We present *CoDA (Communities through Directed Affiliations)*, a method for overlapping community detection that scales to networks with millions of nodes and tens of millions of edges. CoDA exhibits the following three properties: (1) It naturally detects both cohesively connected as well as 2-mode communities. (2) CoDA allows cohesive and 2-mode communities to overlap or be hierarchically nested. (3) CoDA naturally allows for community detection in directed as well as undirected networks.

We develop our community detection method by first presenting a generative model of networks where edges arise from affiliations of nodes to cohesive and 2-mode communities. Then we fit the model to a given network and thus discover communities.

Our model starts with a bipartite affiliation graph [25, 43, 45, 47], where nodes of the underlying network represent one 'layer' of the bipartite graph and communities represent the other. Edges between network-nodes and community-nodes in the affiliation graph represent memberships of nodes to communities. However, our approach has a simple but critical innovation: while memberships of nodes to communities have previously been modeled as undirected, we model the memberships as *directed*.

Though simple on the surface, this modification leads to substantial changes in the modeling capability of affiliation network models. In particular, a directed affiliation between a node and a community models whether the node *sends* or *receives* (or both) links to other members of the community. Directed affiliations allow us to simultaneously model cohesive as well as 2-mode communities. In cohesive communities node affiliations are bidirectional (a node both sends *and* receives links from other members); 2-mode communities are modeled with unidirectional memberships where some members mostly send/create links (*i.e.*, fans) while others mostly receive them (*i.e.*, celebrities).

Having defined the node-community affiliation model we then develop a method to fit the model to a given network. Our model fitting procedure builds on that of the BigCLAM community detection method [45]. Although we solve a more complex problem than BigCLAM (*i.e.*, we find both 2-mode as well as cohesive communities), we employ similar approximation techniques. Until recently, methods for overlapping community detection could only process networks with up to around 10,000 nodes [16]. In contrast, CoDA can easily handle networks that are two orders of magnitude larger: millions of nodes, tens of millions of edges. Moreover, CoDA can be easily parallelized which further increases the scalability.

Present work: Experimental results. We evaluate CoDA on a number of networks from various domains. We consider social, biological, communication, and ecological networks. We test CoDA on networks with explicitly labeled ground-truth communities [31, 44] as well as on networks where communities can be manually examined.

Experiments demonstrate that CoDA's ability to detect 2-mode as well as cohesive communities leads to improved performance over the existing state-of-the-art. For example, when detecting social circles in the Google+ online social network, CoDA gives a relative improvement in accuracy of 36% over Link clustering [2] (28% over MMSB [3], 25% over clique percolation [34] and 21% over DEMON [11]).

More importantly, CoDA facilitates novel discoveries about the community structure of networks. For example, we find that 2-mode communities in foodwebs of predatory relations between organisms correspond to groups of predators who rely on similar groups of prey. Interestingly we find that in scientific paper citation networks, protein-protein interaction networks, as well as web graphs, the majority of detected communities are 2-mode. However, in social networks where edges signify reciprocal friendships, cohesive communities are more frequent. In Twitter or Google+, where relationships are asymmetric, 2-mode communities represent a significant portion of the network (20% in Twitter and 30% in Google+).

Further related work. While there exist a number of different definitions of network communities [14], traditionally, communities have been thought of as densely connected sets of nodes [2, 12, 34, 37]. In contrast, the notion of structural equivalence suggests that nodes with similar connectivity patterns may be considered a community even if they do not link to each other [9, 17, 23]. Our work here builds on both notions of network communities and attempts to resolve them by using a single, unified model.

Detecting communities of densely connected sets of nodes is an extensively researched area [14, 30, 35, 42] with a plethora of different algorithms and heuristics. For example, separate methods have been proposed for detecting communities in undirected networks that are disjoint [4, 13, 21, 38, 40], overlapping [3, 6, 11, 34, 43], or hierarchically nested [2, 44]. On the other hand, detection of 2-mode communities has been much less researched. An exception here is *Trawling* [23], which is a method for extracting 2-mode communities in large directed networks. The critical difference with our work here is that Trawling only identifies *complete* bipartite subgraphs of a given directed network. In contrast, our method is able to identify cohesive as well as bipartite communities in directed as well as undirected networks.

Conceptually CoDA is related to existing work on *block models*, which are in principle capable of detecting cohesive as well as 2-mode communities [3, 16, 18]. Our work differs from such approaches in terms of how communities overlap and are hierarchically nested. We also emphasize the scalability of CoDA compared to these approaches.

CoDA is an example of an affiliation network model [25, 43, 45, 47]. While existing affiliation network models can only model undirected cohesive communities, the crucial difference here is our ability to model *directed* networks and *2-mode* communities.

The rest of the paper is organized as follows. Section 2 defines the affiliation network model and Section 3 discusses the model fitting procedure. We present experimental results in Sections 4 and 5, and conclude in Section 6.

2. DIRECTED COMMUNITY AFFILIATIONS

We start by presenting a stochastic generative model of networks in which the probability of an edge appearing between a pair of nodes depends on the community affiliations of thise nodes. We then develop an efficient model fitting procedure which allows for detecting community affiliations of nodes in a given network.

We describe our model in the context of directed networks and then show how it can straightforwardly be adapted to undirected networks. Our model builds upon BigCLAM, an affiliation model for overlapping network communities [45]. However, whereas BigCLAM focuses on finding only *cohesive* communities in *undirected* networks, our work here aims to find 2-mode communities *as well as* cohesive communities in both directed and undirected networks.

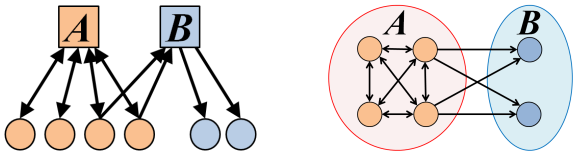
Directed Affiliation Network Model. We begin with the intuition that a desirable model of communities in directed networks should exhibit two properties. First, communities should be modeled not only in terms of their internal connectivity, but also in terms how members connect to non-members. Second, the model should account for asymmetries, *i.e.*, directedness, of edges between nodes. We later demonstrate that accounting for these two properties is important. Perhaps surprisingly, our method gives improved performance even when modeling communities in *undirected* networks. This is due to the fact that when edge directions are not explicit, relationships in the network may still be (implicitly) asymmetric, and identifying such asymmetries leads to improved performance.

We proceed by formulating a simple conceptual model of networks that we refer to as a *Directed Affiliation Network Model*. Our work builds on a family of affiliation network models [8], however, existing affiliation models are typically designed to handle cohesive communities in undirected networks [25, 43, 45, 47]; here we extend such models in order to capture cohesive as well as 2-mode communities in directed as well as undirected networks.

To represent node community memberships, we consider a bipartite affiliation graph where the nodes of the network (bottom layer) connect to communities (top layer) to which they belong (Figure 2(a)). Edges of the underlying network (Figure 2(b)) then arise due to shared community affiliations of nodes.

Consider for a moment an undirected network; when a node belongs to a community in such a network it typically means that the node has (undirected) edges to other members of the community. This type of community affiliation can be modeled using a bipartite graph of nodes and communities where undirected affiliations are formed between nodes and communities [25, 43, 45, 47].

In directed networks, however, we need a richer notion of community affiliation (Figure 2(a)): a node may *create* edges *to* other members of a community, and it also *receive* edges *from* other members of the community, or both. Therefore, we assume that nodes in directed networks can have two "types" of community affiliation: "Outgoing" affiliations from nodes to communities mean that in the network the node *sends* edges to other members of the community. And, "incoming" affiliations from communities to nodes mean that nodes *receive* edges from other community members. We model this using *directed* memberships between nodes and communities: outgoing memberships and incoming memberships.



(a) Node community affiliations (b) Network G

Figure 2: (a) Directed node community affiliation graph. Squares: communities, Circles: nodes of network G. Affiliations from nodes to communities indicate that nodes *create* edges to other members in those communities, while affiliations from communities to nodes indicate that nodes *receive* edges from others. Community A is cohesive, while B is a 2-mode community. (b) Network G corresponding to model in (a).

Formally, we denote a bipartite affiliation graph as $B(V, C, M)$, where V is the set of nodes of the underlying network G, C the set of communities, and M a set of directed edges connecting nodes V and communities C. An outgoing membership edge of node $u \in V$ to community $c \in C$ is denoted as $(u, c) \in M$ and, and an incoming membership is denoted as $(c, u) \in M$.

Now, given the affiliation graph $B(V, C, M)$, we need to specify a process that generates the edges E of the underlying directed network $G(V, E)$. To this end we consider a simple parameterization where we assign a single parameter p_c to every community $c \in C$. The parameter p_c models the probability of a directed edge forming from a member node u with an *outgoing* membership to community c to another member v of c with an *incoming* membership. In other words, we generate a directed edge between a pair of nodes with probability p_c if they are connected in B with a 2-step directed path via community c. Each community c creates edges independently. However, if two nodes are connected by more than one community, duplicate edges are not included in the graph $G(V, E)$.

DEFINITION 1 (DIRECTED AFFILIATION NETWORK MODEL). *Let $B(V, C, M)$ be a directed bipartite graph where V is a set of nodes, C is a set of communities, and M is a set of directed edges between V and C. Also, let $\{p_c\}$ be a set of probabilities for all $c \in C$. Given $B(V, C, M)$ and $\{p_c\}$, the model generates a directed graph $G(V, E)$ by creating a directed edge (u, v) from node $u \in V$ to node $v \in V$ with probability $p(u, v)$:*

$$p(u, v) = 1 - \prod_{k \in C_{uv}} (1 - p_k), \tag{1}$$

where $C_{uv} \subset C$ is a set of communities through which u has a 2-step directed path to v ($C_{uv} = \{c | (u, c), (c, v) \in M\}$). If $C_{uv} = \varnothing$ then we set $p(u, v) = 1/|V|$.

Our Directed Affiliation Network Model and the underlying generated network are illustrated in Figure 3. Directed affiliations are able to explain the overlapping nature of cohesive as well as 2-mode communities. For example, imagine a Twitter network among a community of music fans (A), a community of movie fans (B),

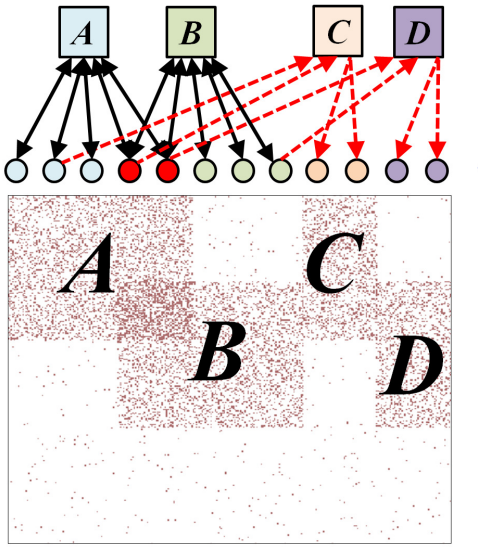


Figure 3: Affiliation graph (top) of the Directed Affiliation Network Model that corresponds to the network adjacency matrix (bottom). It contains two overlapping cohesive (A, B) and two overlapping 2-mode (C, D) communities. Black edges in the affiliation graph denote bidirectional community memberships and red edges denote unidirectional memberships.

a group of famous singers (C), and a group of famous actors (D). Members in communities A and B build bi-directional social relationships inside their respective communities. Some nodes may belong to both communities A and B as they are interested in both movies and music. As for one-directional relationships, we can easily see that music fans would follow singers (C) and movie fans would follow actors (D). Together, these relations would form the adjacency matrix at the bottom of Figure 3. Our model captures this complex community structure very naturally, as shown in the community affiliation graph above the adjacency matrix, where green nodes represent music fans (A), blue nodes are movie fans (B), red nodes are fans of both movies and music, ivory nodes are singers (C), and purple nodes are actors (D). Affiliations between nodes and cohesive communities A and B flow in both directions because members of those communities have reciprocal relationships with each other, whereas fans and celebrities belonging to 2-mode communities C and D have edges flowing in only one direction (fans follow celebrities, celebrities are followed by fans).

More generally, our model has two important advantages over existing approaches [25, 43, 45, 47]: First, CoDA can model natural overlaps between communities. It has been shown that community affiliation models for undirected networks [43] can model community overlaps accurately, which traditional models of overlapping communities fail to capture [2, 3, 34]. The model also captures realistic community overlaps because its modeling power *generalizes* that of other community affiliation models for undirected networks, *i.e.*, CoDA can model overlaps between cohesive communities *in addition to* 2-mode communities. The second advantage of our model is its ability to model 2-mode communities. By modeling such communities, we can better capture the interaction between groups of nodes. This is a significant improvement over current methods that model only interactions *within* communities.

3. COMMUNITY DETECTION

Given an unlabeled, directed network $G(V, E)$, our goal is to identify cohesive as well as 2-mode communities. We achieve this by fitting our *Directed Affiliation Network Model* to $G(V, E)$, *i.e.*, by finding an affiliation graph B and parameters $\{p_c\}$ that maximize the data likelihood. For now, we assume that the number of communities K is given; we will later discuss how to automatically determine K. We aim to solve the following Maximum Likelihood Estimation problem:

$$\underset{P,\{p_c\}}{\operatorname{argmax}} \sum_{(u,v)\in E} \log p(u,v) + \sum_{(u,v)\notin E} \log(1-p(u,v)), \quad (2)$$

where the edge probability $p(u, v)$ is defined in Eq. 1.

Eq. 2 leads to a challenging optimization problem. Specifically, it involves a combinatorial search over all possible affiliation graphs B [43]. Therefore, we develop an approximate algorithm for optimizing Eq. 2. We achive this by relaxing the original problem by changing binary memberships into real-valued memberships.

We build on the intuition from the BigCLAM [45] optimization procedure and begin by introducing variables to represent the memberships of the nodes. As noted earlier, we distinguish nodes' incoming memberships and outgoing memberships. In particular, let M_{uc} indicate whether the node u belongs to community c with an outgoing membership, and L_{vc} indicate whether node v has an incoming membership for c. Now Eq. 1 can be represented as:

$$p(u,v) = 1 - \prod_{c\in C_{uv}} (1-p_c) = 1 - \prod_c (1-p_c)^{M_{uc}L_{vc}},$$

By applying the change of variables $1 - p_c = \exp(-\alpha_c)$ with $\alpha_c \geq 0$, the equation becomes linear in M, L, and α_c:

$$p(u,v) = 1 - \exp(-\sum_c M_{uc}\alpha_c L_{vc}).$$

We then further simplify the equation by letting $\tilde{M}_{uc} = \sqrt{\alpha_c} M_{uc}$ and $\tilde{L}_{vc} = \sqrt{\alpha_c} L_{vc}$.

$$p(u,v) = 1 - \exp(-\sum_c \tilde{M}_{uc}\tilde{L}_{vc}).$$

So far, we have not used any approximations and the problem is still combinatorial since the variables remain restricted: $\tilde{M}_{uc} \in \{\sqrt{\alpha_c}, 0\}$ and $\tilde{L}_{vc} \in \{\sqrt{\alpha_c}, 0\}$.

However, note that we can interpret $\tilde{M}_{uc}$ as the strength of the membership of node u to community c. Thus the condition $\tilde{M}_{uc} \in \{\sqrt{\alpha_c}, 0\}$ simply means that if node u belongs to c, it would be connected to other member nodes in c with the factor $\sqrt{\alpha_c}$, which determines p_c. The same argument also applies to $\tilde{L}_{vc}$.

Now we replace $\tilde{M}_{uc}$ and $\tilde{L}_{vc}$ with *nonnegative continuous* valued memberships F_{uc} and H_{vc}, respectively. The advantage here is that now each node can pick the "strength" of its membership to a given community: A high value of F_{uc} means that the node u has many outgoing edges towards other members of c, while high H_{vc} means that node v has many incoming edges from other members of c. Now we ca write:

$$p(u,v) = 1 - \exp(-F_u H_v^T).$$

And we transformed Eq. 2 into a continuous optimization problem:

$$\{\hat{F}, \hat{H}\} = \underset{F,H\geq 0}{\operatorname{argmax}}\, l(F,H) \quad (3)$$

where

$$l(F,H) = \sum_{(u,v)\in E} \log(1-\exp(-F_u H_v^T)) - \sum_{(u,v)\notin E} F_u H_v^T.$$

In other words, in order to detect network communities we fit our model by estimating non-negative affiliation matrices $\hat{F}, \hat{H} \in \mathbb{R}^{N \times K}$ that maximize the likelihood $l(F, H) = \log P(G|F, H)$.

Solving the optimization problem. To solve the problem in Eq. 3, we adopt a block coordinate ascent approach: We update F_u for each u with H fixed and update H_v for each v with F fixed, *i.e.*, we update either incoming or outgoing memberships of one node while fixing the other type of memberships. This approach has the advantage that each subproblem of updating F_u and H_v is convex. For brevity we describe only how to update F_u. Updating H_v is analogous. For each u we solve:

$$\underset{F_{uc} \geq 0}{\operatorname{argmax}}\, l(F_u), \tag{4}$$

where

$$l(F_u) = \sum_{v \in \mathcal{N}(u)} \log(1 - \exp(-F_u H_v^T)) - \sum_{v \notin \mathcal{N}(u)} F_u H_v^T,$$

where $\mathcal{N}(u)$ is a set of neighbors of u. To solve this convex problem, we use projected gradient ascent with the following gradient:

$$\nabla l(F_u) = \sum_{v \in \mathcal{N}(u)} H_v \frac{\exp(-F_u H_v^T)}{1 - \exp(-F_u H_v^T)} - \sum_{v \notin \mathcal{N}(u)} H_v$$

We compute the step size using backtracking line search. After each update, we project F_u into a space of nonnegative vectors by setting $F_{uc} = \max(F_{uc}, 0)$.

Naive computation of $\nabla l(F_u)$ takes time $O(|V|)$. However, we reduce the computational complexity to the *degree* of u, $O(|\mathcal{N}(u)|)$, which significantly increases the scalability of our approach. We achieve this by computing the second term $\sum_{v \notin \mathcal{N}(u)} H_v$ in $O(|\mathcal{N}(u)|)$ by storing/caching $\sum_v H_v$:

$$\sum_{v \notin \mathcal{N}(u)} H_v = (\sum_v H_v - H_u - \sum_{v \in \mathcal{N}(u)} H_v).$$

Given that real-world networks are extremely sparse ($|\mathcal{N}(u)| \ll N$), we can update F_u for a single node u in *near-constant* time. The update rule for H_v can be similarly derived and takes near-constant time $O(|\mathcal{N}(v)|)$. In practice, we iteratively update F_u, H_u for each u and stop iterating once the likelihood does not increase (by 0.01%) after we update F_u, H_u for all u.

Determining community affiliations of nodes. From the real-valued $\hat{F}, \hat{H}$ that we estimate, we want to determine "hard" community affiliations of nodes. We achieve this by thresholding F_{uc} and H_{uc} with a constant δ, *i.e.*, we regard u has an outgoing membership to community c if $F_{uc} \geq \delta$, and an incoming membership from c if $H_{uc} \geq \delta$.

We choose the value of δ so that every pair of members in community c has edge probability higher than the background edge probability $1/|V|$ (see Eq. 1):

$$\frac{1}{|V|} \leq 1 - \exp(-\delta^2)$$

This inequality leads to $\delta = \sqrt{-\log(1 - 1/|V|)}$. We note that we also experimented with other values of δ and found that this choice for δ works well in practice.

Algorithm initialization. To initialize F, H, we employ *locally minimal neighborhoods*, which provide good seed-sets for community discovery [15]. A neighborhood $N(u)$ of a node u is a set consisting of the node u and its neighbors, and $N(u)$ is said to be "locally minimal" if $N(u)$ has lower conductance score than $N(v)$ for any other neighbor v of u [15]. For a node u' belonging to such a locally minimal neighborhood k, we initialize $F_{u'k} = 1$ if u' has an outgoing edge (or $F_{u'k} = 0$ otherwise), and set $H_{u'k} = 1$ if u' has an incoming edge (or $H_{u'k} = 0$ otherwise).

Choosing the number of communities. To automatically determine the number of communities K, we follow the approach proposed in [3]. We divide all node pairs into 80% training and 20% test set. Varying K, we fit CoDA with K communities on the training pairs and measure the likelihood for the test pairs. We then select K with the highest test set likelihood. For a small networks with fewer than 100 edges, we find that a different criterion works better in practice. Here we choose K so as to achieve the smallest value of the Bayesian Information Criterion:

$$BIC(K) = -2l(\hat{F}, \hat{H}) + NK \log |E|.$$

Parallelization and implementation details. Our approach also naturally allows for *parallelization*, which further increases scalability of CoDA. When updating F_u for each node u (Eq. 4), we observe that each subproblem is *separable* since all other variables in Eq. 4 (H) remain fixed. That is, updating the value of F_u for a specific node u does not affect updates of F_v for all other nodes v. In the parallelized version of CoDA, we solve Eq. 4 for multiple nodes in parallel. This parallelization does not affect the final result of the method. Updating H_u for each node u can be parallelized in the same way. As we show in Section 4, parallelization on a single shared memory machine boosts the speed of CoDA by a factor of 20 (the number of threads) used when analyzing a 300,000 node network. Last, we also experimented with other optimization techniques such as the cyclic coordinate descent method (CCD) [19] which optimizes F_{uc} for each u and each c by Newton's method, but we found that block coordinate ascent converges the fastest.

A parallel C++ implementation of CoDA is publicly available at `http://snap.stanford.edu`.

CoDA for undirected networks. So far, we have discussed CoDA under the context of directed networks. However, CoDA can easily be applied to undirected networks as well. We make a simple observation: undirected networks model symmetric relationships and thus an undirected relationship is equivalent to two directed relationships, one each way. Thus, given an undirected network, we simply convert the network into a directed one by regarding every edge as reciprocal, and then apply CoDA to detect communities.

Now, CoDA will easily detect cohesive communities in this converted network as edges in cohesive communities are reciprocal. Detecting 2-mode communities is also simple. Consider the case where we are given an undirected 2-mode community X where nodes in group A are connected to nodes in group B. Once we convert X into a directed network with reciprocal edges between A and B, CoDA will estimate two 2-mode communities from this community X: $\hat{X}_1$ for edges from A to B, and $\hat{X}_2$ for edges from B to A. Thus, CoDA is able to correctly discover X, with the caveat that it discovers it twice (both $\hat{X}_1$ and $\hat{X}_2$ correspond to X).

4. EXPERIMENTS

We evaluate the performance of CoDA and compare it to state-of-the-art community detection methods on a range of directed as well as undirected networks. We measure the quality of community detection by computing the detection accuracy based on gold-standard ground-truth communities. We also evaluate the scalability of the methods by measuring runtime as network size increases.

4.1 Dataset Description

We begin by briefly describing the networks that we consider in this study. Overall, we consider 5 undirected and 9 directed net-

Method	F_1 score Google+	Twitter	Facebook	Jaccard similarity Google+	Twitter	Facebook	Average
MMSB [3]	0.324 (0.033)	0.262 (0.005)	0.374 (0.042)	0.214 (0.026)	0.169 (0.004)	0.266 (0.036)	0.268
Clique percolation [34]	0.331 (0.036)	0.246 (0.006)	0.429 (0.051)	0.240 (0.032)	0.163 (0.005)	0.342 (0.050)	0.292
Link clustering [2]	0.304 (0.016)	0.334 (0.003)	0.372 (0.027)	0.226 (0.016)	**0.238 (0.003)**[2]	0.275 (0.024)	0.291
BigCLAM [43]	0.324 (0.017)	0.344 (0.005)	0.442 (0.042)	0.217 (0.014)	0.234 (0.004)	0.325 (0.038)	0.315
DEMON [11]	0.343 (0.029)	0.308 (0.005)	0.418 (0.046)	0.255 (0.027)	0.210 (0.005)	0.311 (0.041)	0.307
NMF [29]	0.333 (0.019)	0.318 (0.004)	0.406 (0.038)	0.242 (0.026)	0.221 (0.004)	0.301 (0.050)	0.303
CoDA, undirected	**0.414 (0.027)**[1]	**0.348 (0.005)**[2]	**0.470 (0.042)**[1]	**0.314 (0.026)**[1]	0.237 (0.004)	**0.357 (0.039)**[1]	**0.357**[2]
CoDA, directed	**0.406 (0.025)**[2]	**0.363 (0.005)**[1]	**0.470 (0.042)**[1]	**0.314 (0.024)**[1]	**0.250 (0.004)**[1]	**0.357 (0.039)**[1]	**0.360**[1]

Table 2: Performance on Facebook, Google+, and Twitter. Higher is better. Standard errors are shown in parentheses. The best and second best methods are annotated as '1' and '2'.

Dataset	Directed	N	E	C	S	A
Google+	✓	250,469	30,230,905	437	143.51	0.25
Twitter	✓	125,120	2,248,406	3,140	15.54	0.39
Facebook	✗	4,089	170,174	193	28,76	1.36
Enron	✓	45,266	185,172	4,572	63.93	6.46
LiveJournal	✗	3,997,962	34,681,189	287,512	22.31	1.59
Youtube	✗	1,134,890	2,987,624	8,385	13.50	0.10

Table 1: Dataset statistics. Directed: Yes/no, N: number of nodes, E: number of edges, C: number of ground-truth communities, S: average ground-truth community size, A: ground-truth community memberships per node. Further datasets used in this study are described in Table 5.

works from a wide spectrum of domains. We consider social, communication, information, biological and ecological networks.[1]

Networks with ground-truth communities. For the experiments in this section, we consider a subset of 6 publicly available networks where we have explicit *ground-truth* memberships of nodes to communities [44]. The availability of ground-truth allows us to quantify the quality of community detection methods *quantitatively*. Table 1 shows the statistics of the networks and the ground-truth communities. The networks come from three different domains: The first three networks are the collection of ego-networks from online social networks of Facebook, Twitter and Google+ [31], the Enron email communication network [22], and LiveJournal and Youtube social networks [33]. We describe the nature of ground-truth communities in each of these datasets in more detail later.

4.2 Experimental Setup

Baselines. For comparison we consider the following baseline methods: *MMSB* (Mixed Membership Stochastic Blockmodels) [3], which can detect both cohesive and 2-mode communities in undirected networks and is extremely slow; *Clique Percolation*, [34] *Link Clustering* [2], *BigCLAM* [43, 45] are state of the art overlapping cohesive community detection techniques for undirected networks; *DEMON* [11] is a scalable local community detection method for directed networks; *NMF* [29] is a state-of-the-art non-negative matrix factorization approach which can be used for directed networks. We use publicly available implementations of each of the methods.

Some methods require input parameters. MMSB and NMF requires the number of communities K. We use the Bayes information criterion suggested by the authors [3] to choose K. DEMON requires ε, the threshold value for merging two communities. As there exists no standard criterion for ε, we set ε so that DEMON detects the same number of communities as CoDA does.

[1]We use the publicly available data from the Stanford Large Network Collection: `http://snap.stanford.edu`.

Last, we note that the above baselines represent the current state-of-the-art in community detection. However, we also considered other baselines, including those that make use of node features [20], network topology [39], or both [5, 31]; however experiments demonstrate that none of these alternatives outperforms CoDA.

Evaluation. To evaluate the performance of the above methods we quantify the degree of correspondence between the ground-truth and the detected communities. To compare a set of ground-truth communities C^* to a set of predicted communities C, we adopt an evaluation procedure previously used in [43, 45], where every detected (ground-truth) community is matched with its most similar ground-truth (detected) counterpart community:

$$\frac{1}{2|C^*|}\sum_{C_i^*\in C^*}\max_{C_j\in C}\delta(C_i^*,C_j)+\frac{1}{2|C|}\sum_{C_j\in C}\max_{C_i^*\in C^*}\delta(C_i^*,C_j),$$

where $\delta(C_i^*, C_j)$ is some measure of the similarity between the communities C_i^* and C_j. We consider two standard measures of the similarity between sets, namely the F_1 score and the Jaccard similarity. Thus, we obtain a value between 0 and 1, where 1 indicates perfect recovery.

4.3 Detecting Social Circles

First we consider the problem of discovering users' social circles [31]. Circles (or 'lists' in Facebook and Twitter) give users a means of categorizing their immediate neighbors, or in the case of directed networks, the users whom they follow. Thus the problem of automatically identifying users' social circles can be posed as a community detection problem on each user's ego-network [31].

In Table 2 we evaluate the performance of CoDA and baselines on social circle detection. Across all three datasets and both evaluation metrics, CoDA (the last row) is the best or second-best performer. On average, CoDA outperforms MMSB by 34%, Clique percolation by 23%, Link clustering by 24%, BigCLAM by 14%, DEMON by 17%, and NMF by 19%.

The 3 data sets possess very different reasons for community (*i.e.*, social circle) formation: Facebook is an undirected network and in Facebook circles are driven by dense mutual friendships among users with homogeneous backgrounds [31]; therefore, we would expect cohesive communities in Facebook. Google+ and Twitter are directed networks and as such circles are not necessarily based on friendship, because edges in these networks denote *follower* relationships: The fraction of reciprocated edges is only 29% in Google+ and 54% in Twitter. For example, a social circle in Twitter might consist of authors who publish in the same genre, or candidates in the same election. As we will see later in Section 5, many social circles in Google+ and Twitter follow such 2-mode structure.

Regardless of very different nature of the data sets, CoDA is the best performing method in each of them. This result means that CoDA recovers 2-mode circles in Google+ or Twitter *as well as*

Method	F_1 score	Jaccard similarity	Average
MMSB	N/A	N/A	N/A
Clique percolation	N/A	N/A	N/A
Link clustering	0.195	0.294	0.245
BigCLAM	0.478	0.358	0.418
DEMON	0.464	0.350	0.407
NMF	N/A	N/A	N/A
CoDA, undirected	0.538	0.431	0.485
CoDA, directed	0.617	0.516	0.567

Table 3: Performance of recipient discovery on the Enron network. Algorithms that do not scale to the size of the dataset are labeled as "N/A".

cohesive circles in Facebook, *i.e.*, CoDA can detect *both* kinds of communities more accurately than the baselines.

Directed vs. undirected networks. To further examine the performance out method on directed and undirected networks we perform an experiment with the goal of understanding whether CoDA is still able to recover 2-mode communities even when edge directions are dropped and networks are considered as undirected. To test this, we convert the directed networks of Twitter and Google+ into undirected by removing the edge directions. Then we apply CoDA (CoDA, undirected, the second to last row in Table 2). Surprisingly, CoDA achieves similar performance even without explicit edge directions in the network. Based on this evidence we conclude CoDA is capable of accurately finding 2-mode communities even in undirected networks.

4.4 Discovering Recipient Lists in Email Networks

We also define a task of automatically discovering recipient lists in the the email communication network. The idea is that such lists exhibit a distinct structural pattern in the network as the recipient lists may have 2-mode community structure as a set of users who receive the same email may not necessarily email each other [32].

We consider all Enron emails [22] with 20 or more recipients. This gives us a set of 4,572 unique recipient lists in the Enron dataset, which we treat as ground-truth communities (Table 1). Now we are given an unlabeled directed Enron email communication network, where an edge $i \rightarrow j$ means that i sent at least one mail to j, and the goal is to discover email recipient lists.

We then apply CoDA as well as the baselines to this network and in Table 3 we measure how accurately the communities detected by CoDA correspond to these ground-truth email recipient lists. We report both the F_1 score and Jaccard similarity (for methods that do not scale to networks of this size, we report N/A). Table 3 shows that CoDA outperforms other methods by a significant margin. CoDA outperforms Link clustering by 131%, DEMON by 39%, and BigCLAM by 36%.

4.5 Experiments on Large Networks

Last, we also examine two real-world social networks with millions of nodes in which nodes explicitly declare their community memberships [44]. We consider the LiveJournal and Youtube social networks, and regard user-created groups as ground-truth communities. We ignore groups containing fewer than 10 nodes, yielding 71,093 communities in LiveJournal and 2,078 in Youtube.

Of the baselines previously mentioned, only BigCLAM could scale to both networks and DEMON could scale to the Youtube network. Therefore, we also consider two large-scale graph partitioning methods as baselines for this experiment: Metis [21] and Graclus [13]. For all methods we set the number of communities K to be the number of ground-truth communities.

Method	Relative F_1 score LiveJournal	Relative F_1 score Youtube	Absolute F_1 score LiveJournal	Absolute F_1 score Youtube
Metis	100%	200%	0.12	0.028
Graclus	100%	185.7%	0.12	0.026
BigCLAM	121.0 %	278.1 %	0.14	0.039
DEMON	N/A	100%	N/A	0.014
CoDA	129.4%	307.1%	0.15	0.043

Table 4: Relative accuracy (compared to the worst performing method) of detected communities on large scale social networks.

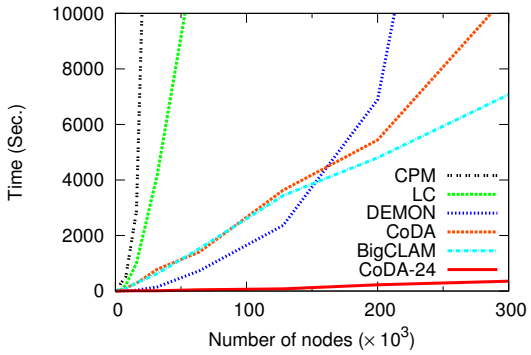


Figure 4: Algorithm runtime.

Table 4 shows the results. For this experiment we focus on the score relative to that of the worst-performing baseline in each network (so that the worst-performing baseline has a score of 100%). We compute the relative score because the networks are only partially labeled and the overall performance is thus artificially low (as methods discover many unlabeled communities). We find that CoDA outperforms its nearest competitor by 8.4% on LiveJournal and 29% on Youtube.

4.6 Scalability

Last, we evaluate the scalability of CoDA by measuring its running time on synthetic networks with increasing size. We generate synthetic networks using the Forest fire model [27] with the forward and backward probabilities set to 0.36 and 0.32, respectively. Since CoDA is easily parallelizable as described in Section 3, we also consider a singe machine parallel implementation running with 24 threads (CoDA-24).

Scalability results are shown in Figure 4. Link Clustering and Clique Percolation scale to networks of at most a few thousand nodes. DEMON is a fast and scalable overlapping community detection method. DMEON tends to be faster than CoDA (single-threaded implementation) for networks up to 100,000 nodes, however, once the network becomes larger, CoDA becomes much faster.

When comparing single-threaded implementations we also note that BigCLAM is the fastest method in our experiments. However, we note that CoDA takes only 30% more time than BigCLAM while it is also solving a more complicated problem, namely detecting cohesive as well as 2-mode communities.

Last, we also measure a parallelized version of CoDA (CoDA-24). Using 24 threads on a single machine, we achieve nearly 24x speedup. Ultimately, CoDA takes just 6 minutes to process a 300,000 node network.

Dataset	Directed	N	E	C	S	A
PPI-Y2H	✗	1,647	2,518	40	90.75	2.20
PPI-LC	✗	1,213	2,556	40	42.08	1.39
web-Stanford	✓	281k	2,312k	19k	70.63	4.59
web-Google	✓	875k	5,105k	39k	41.79	1.86
cit-HepTh	✓	27k	353k	2,000	70.00	5.04
cit-HepPh	✓	34k	422k	4,976	51.52	7.42
Florida Bay	✓	121	1,745	6	45.33	2.25
Chesapeake	✓	33	72	5	9.20	1.39

Table 5: Dataset statistics. Directed: Whether the network is directed or not, N: number of nodes, E: number of edges, C: number of detected communities, S: average size of detected communities, A: community memberships per node.

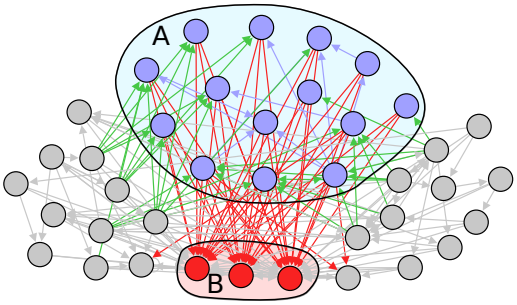


Figure 5: Two detected communities in a Foodweb (Chesapeake Bay). Among other communities, CoDA identifies sets of nodes with similar predators (A, blue nodes) and with similar prey (B, red nodes), both of which have low internal connectivity.

5. COMMUNITY DISCOVERY

So far we have demonstrated that CoDA can reliably detect both cohesive and 2-mode communities in directed as well as undirected networks. In the following section, we shall demonstrate that 2-mode communities take an important role in networks. We shall use CoDA to perform a qualitative study of various networks in order to determine the extent to which community structures vary across real-world networks from various domains.

Network data. In addition to the datasets already introduced, we also analyze biological networks, foodwebs, web graphs, and citation networks (Table 5). For biological networks, we consider the protein-protein interaction network of *Saccharomyces cerevisiae*: yeast two-hybrid (PPI-Y2H) and literature-curated (PPI-LC) [2]. We also consider the Chesapeake and Florida Bay foodwebs [41], the web graph of Stanford University web pages (web-Stanford), the web graph released by Google in 2002 (web-Google) [28], and the arXiv citation networks from high-energy physics phenomenology (cit-HepPh) and theory (cit-HepTh) [27] all available from `http://snap.stanford.edu`.

5.1 Biological and Foodweb Communities

We first present 2-mode communities in foodwebs, where nodes represent organisms and an edge from a node u to v means that u is preyed upon by v. We apply CoDA on the Chesapeake Bay foodweb network shown in Figure 5, and display an induced subgraph of detected 2-mode communities in Figure 6.

In foodweb networks, we find 2-mode communities of groups of predators who rely on similar groups of prey (Figure 6). The blue 2-mode community (B-D) represents predators and prey in the the Chesapeake Bay sands: *nereis*, *macoma spp.*, and *mya arenaria* (in B) are small, sand-dwelling clams and worms that are fed on by fish (in D). Alternately, the red community (A-C) shows predator-

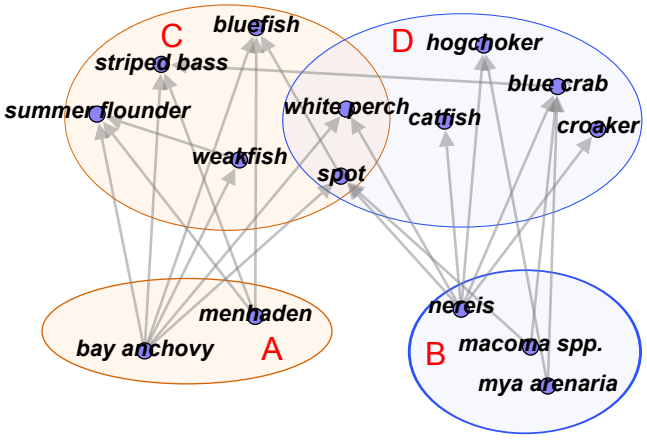


Figure 6: Examples of overlapping 2-Mode communities detected by our method in the Chesapeake bay foodweb network. See main text for the explanation of community structure.

prey relationships among fish: small fish (A) are eaten by bigger fish (C). CoDA also discovers the *overlap* between two predator groups where *white perch* and *spot* prey on both fish and clams.

CoDA also allows us to gain insights into biological PPI networks. Interestingly, CoDA discovers many 2-mode communities in the undirected protein-protein interaction network determined by yeast two-hybrid screening (PPI-Y2H). For example, Figure 8 displays the induced subgraph of two communities that CoDA detects. 2-mode communities detected by CoDA clearly reveal the interaction between different protein groups. For example, proteins in group C of Figure 8 heavily interact with proteins in group A, even though these proteins do not interact within the same group (with A or within C).

To further analyze the role of these communities, we used gene ontologies to identify relevant terms/functions of proteins in A, B, C, and D using the GO Term Finder [7]. The proteins in the large groups (C, D) are generally associated with catalytic activity and ion binding (p-value $\sim 10^{-4}$).

However, these proteins are regulated by different protein groups (A, B) which have different functions. Proteins in A (*e.g.*, YLR347C and YNL189W) are protein transporters, whereas proteins in B (*e.g.*, YLR291C) are regulators. Perhaps more interestingly, YPL070W belongs to both A and B and regulates both C and D. However, its role is not yet known. But based on known functions of proteins in groups A and B we can extrapolate the function of YPL070W. This example shows how network analysis and community detection in particular can provide research directions for experimental biology [10].

5.2 2-mode vs. Cohesive Communities

Since CoDA can detect both cohesive and 2-mode communities, we can use it to measure the extent to which real network data exhibits cohesive and 2-mode behavior. This analysis allows us to characterize the mesoscale structure of real-world networks as the proportion of 2-mode versus cohesive communities can be used to gain further insights into community structure of networks.

Experimental setup. For this experiment, we consider 12 networks from 6 domains in order to characterize their different community structures. We consider ego networks (Twitter, Google+) and social networks (LiveJournal, Youtube) from Section 4. We also include 8 networks from 4 different domains: Biological networks,

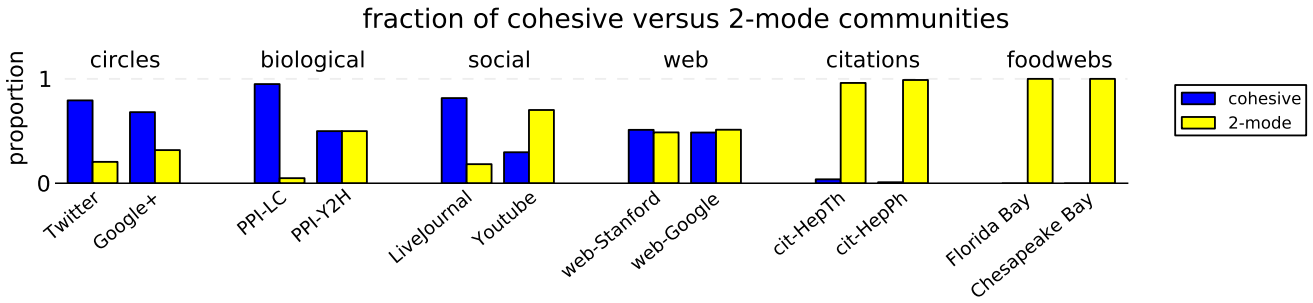


Figure 7: Fraction of 2-Mode communities and cohesive communities in six different types of networks.

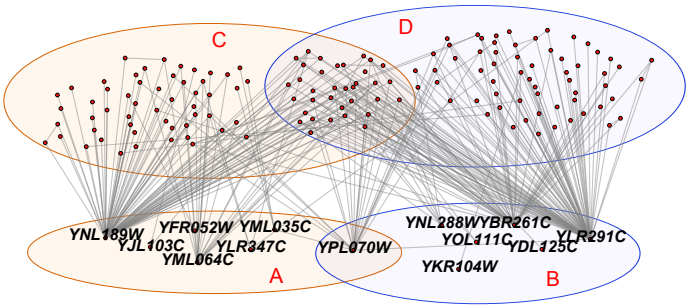


Figure 8: Overlapping 2-Mode communities detected by our method in a Protein-Protein interaction network. See main text for the explanation of community structure.

web graphs, foodwebs, and citation networks among research papers from Table 5.

To classify whether a detected community is 2-mode or cohesive, we measure the Jaccard similarity $J(c) = \frac{|O(c) \cap I(c)|}{|O(c) \cup I(c)|}$ between the set of member nodes with outgoing memberships $O(c)$, and the set of member nodes with incoming memberships $I(c)$. In a completely cohesive community, this Jaccard similarity is 1 because two sets of members are identical, whereas it is 0 in a completely 2-mode community. We regard a community c as 2-mode if $J(c)$ is lower than some threshold γ or as cohesive otherwise ($J(c) \geq \gamma$). We use $\gamma = 0.2$ as this setting gives the most interpretable results.

Experimental results. Figure 7 shows the fraction of 2-mode and cohesive communities in 12 networks described above. Ego networks (Twitter and Google+) exhibit a relatively high fraction of cohesive communities and as noted earlier Facebook ego networks (not shown) have an even higher fraction (over 95%) of cohesive communities. This result is in line with [11] where the authors show that Facebook ego networks can be easily divided into cohesive communities. However, it is important to note that a significant fraction of Twitter (20%) and Google+ (30%) communities exhibit 2-mode structure.

Literature-curated protein-protein interaction networks (PPI-LC) practically have only cohesive communities (and no 2-mode). On the other hand, in PPI networks generated based on yeast two-hybrid screening (PPI-Y2H) about 50% of the communities are 2-mode. This difference is interesting and confirms a previous study of PPI networks [46], which provided the following explanation: Edges of PPI-LC are extracted from scientific papers that report experimentally validated interactions. However, current biological experiments have mainly been guided by research on cohesive communities and thus it seems as though most interactions that have been explored take place in "cohesive" communities [46]. On the other hand, the PPI-Y2H network is created by a noisy automatic process and more faithfully represents the interaction network. In this case many 2-mode communities emerge [46].

In social networks we also find interesting results. In LiveJournal, communities are more cohesive, which can be explained by the fact that edges in LiveJournal indicate "friendships" (*i.e.*, sharing private blog content). On the other hand, Youtube communities are predominantly 2-mode. Youtube differs from other social networks in one important way: Edges in Youtube are essentially "subscriptions" for content rather than mutual friendships; consequently, high degree nodes tend to connect to low degree nodes [33].

Web graphs are of interest because Kumar et al. [23] used the existence of 2-mode communities as indicators or signatures for cohesive communities. Our results nicely suggest the co-existence of cohesive communities and 2-mode communities by showing that web graphs have an equal proportion of 2-mode and cohesive communities.

Finally, foodwebs as well as citation networks consist almost entirely of 2-mode communities. These results are natural as reciprocal and cohesive relationships are extremely unlikely in these networks. In foodwebs, for example, few species prey upon each other. Citation networks are directed acyclic graphs and reciprocal citation is impossible by definition. Intuitively, cohesive communities in directed networks contain some number of bidirectional edges among their members, therefore a lack of such reciprocal edges naturally leads to the dominance of 2-mode communities, as we observe in Figure 6.

6. CONCLUSION

An accurate notion of a *community* is critical when studying the mesoscale structure of networks. Traditional models consider 'communities' to be sets of densely connected nodes. In addition, here we also consider *2-mode* communities, which are groups of nodes who may not link to each other but link in a coordinate way to the other nodes in the network.

We have presented CoDA, a community detection method which naturally detects both densely connected and 2-mode communities. CoDA can capture overlapping and hierarchical structure among communities, and handles both directed and undirected networks. Our experimental findings reveal that CoDA outperforms the current state-of-the-art in detecting ground-truth communities. Moreover, CoDA also reveals how 2-mode and cohesive communities co-exist in real networks.

The versatility of CoDA to detect both cohesive and 2-mode communities accurately in directed and undirected networks raises many interesting avenues of future work. For example, understanding the interaction between 2-mode communities and cohesive communities is a fruitful direction. Inferring the role of nodes from their community affiliations would also be useful. Another idea is to extend CoDA to find important nodes in each community. This could be achieved by the fact that CoDA estimates real-valued member-

ship strengths (F_{uc} and H_{uc}) of each node to each community. From the values of F_{uc} and H_{uc} for node u and community c, we could determine which nodes are most important and have the "heaviest" membership to a given community c.

Acknowledgements. This research has been supported in part by NSF IIS-1016909, CNS-1010921, CAREER IIS-1149837, IIS-1159679, ARO MURI, DARPA GRAPHS, ARL AHPCRC, Okawa Foundation, PayPal, Docomo, Boeing, Allyes, Volkswagen, Intel, Alfred P. Sloan Fellowship, and the Microsoft Faculty Fellowship.

7. REFERENCES

[1] L. Adamic and N. Glance. The political blogosphere and the 2004 U.S. election: divided they blog. In *LinkKDD '05*, 2005.

[2] Y.-Y. Ahn, J. Bagrow, and S. Lehmann. Link communities reveal multi-scale complexity in networks. *Nature*, 2010.

[3] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *JMLR*, 2007.

[4] R. Andersen and K. Lang. Communities from seed sets. In *WWW '06*, 2006.

[5] R. Balasubramanyan and W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *SDM '11*, 2011.

[6] B. Ball, B. Karrer, and M. Newman. Efficient and principled method for detecting communities in networks. In *Phys. Rev. E*, 2011.

[7] E. Boyle et al. GO::TermFinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 2004.

[8] L. Breiger. The duality of persons and groups. *Social Forces*, 1974.

[9] R. Burt. Cohesion versus structural equivalence as a basis for network subgroups. *Sociological Methods and Research*, 1978.

[10] D. Carney, B. Davies, and B. Horazdovsky. Vps9 domain-containing proteins: activators of Rab5 GTPases from yeast to neurons. *Trends in Cell Biology*, 2006.

[11] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Demon: a local-first discovery method for overlapping communities. In *KDD '12*, 2012.

[12] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *J. of Stat. Mech.: Theory and Experiment*, 2005.

[13] I. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE PAMI*, 2007.

[14] S. Fortunato. Community detection in graphs. *Physics Reports*, 2010.

[15] D. Gleich and Seshadhri. Neighborhoods are good communities. In *KDD '12*, 2012.

[16] P. Gopalan, D. Mimno, S. Gerrish, M. Freedman, and D. Blei. Scalable inference of overlapping communities. In *NIPS '12*, 2012.

[17] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. Rolx: structural role extraction & mining in large graphs. In *KDD*, 2012.

[18] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 1983.

[19] C.-J. Hsieh and I. S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *KDD '11*, 2011.

[20] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 1967.

[21] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. of Parallel and Distributed Computing*, 1998.

[22] B. Klimt and Y. Yang. Introducing the enron corpus. In *CEAS '04*, 2004.

[23] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 1999.

[24] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW '10*, 2010.

[25] S. Lattanzi and D. Sivakumar. Affiliation networks. In *STOC '09*, 2009.

[26] E. Leicht and M. E. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 2008.

[27] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05*, 2005.

[28] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 2009.

[29] C-J. Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, 2007.

[30] F. Malliaros and M. Vazirgiannis. Clustering and Community Detection in Directed Networks: A Survey. *Physics Reports*, 2013.

[31] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS '12*, 2012.

[32] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *JAIR*, 2007.

[33] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07*, 2007.

[34] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 2005.

[35] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos. Community detection in social media. *DMKD*, 2011.

[36] S. Pinkert, J. Schultz, and J. Reichardt. Protein interaction networks—more than mere modules. *PLoS CompBio*, 2010.

[37] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *PNAS*, 2004.

[38] M. Rosvall and C. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *PNAS*, 2007.

[39] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 2008.

[40] V. Satuluri and S. Parthasarathy. Scalable Graph Clustering using Stochastic Flows: Applications to Community Discovery. *KDD '09*, 2009.

[41] R. Ulanowicz, C. Bondavalli, and M. Egnotovich. Network analysis of trophic dynamics in south florida ecosystem, fy 97: The florida bay ecosystem. *Ref. CBL98-123. Chesapeake Biological Laboratory*, 1998.

[42] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Surveys*, 2013.

[43] J. Yang and J. Leskovec. Community-affiliation network model for overlapping community detection. In *ICDM '12*, 2012.

[44] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *ICDM '12*, 2012.

[45] J. Yang and J. Leskovec. Overlapping community detection at scale: A non-negative factorization approach. In *WSDM '13*, 2013.

[46] H. Yu, P. Braun, M. A. Yildirim, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane-Kishikawa, F. Gebreab, N. Li, N. Simonis, and et al. High-quality binary protein interaction map of the yeast interactome network. *Science*, 2008.

[47] E. Zheleva, H. Sharara, and L. Getoor. Co-evolution of social and affiliation networks. In *KDD '09*, 2009.