Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach

Jaewon Yang
Stanford University
crucis@stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

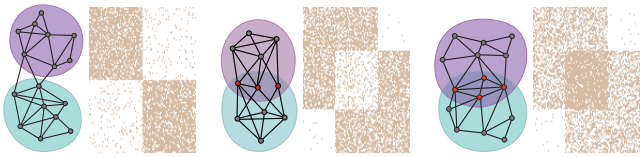
Network communities represent basic structures for understanding the organization of real-world networks. A community (also referred to as a module or a cluster) is typically thought of as a group of nodes with more connections amongst its members than between its members and the remainder of the network. Communities in networks also overlap as nodes belong to multiple clusters at once. Due to the difficulties in evaluating the detected communities and the lack of scalable algorithms, the task of overlapping community detection in large networks largely remains an open problem.

In this paper we present BIGCLAM (Cluster Affiliation Model for Big Networks), an overlapping community detection method that scales to large networks of millions of nodes and edges. We build on a novel observation that overlaps between communities are densely connected. This is in sharp contrast with present community detection methods which implicitly assume that overlaps between communities are sparsely connected and thus cannot properly extract overlapping communities in networks. In this paper, we develop a model-based community detection algorithm that can detect densely overlapping, hierarchically nested as well as non-overlapping communities in massive networks. We evaluate our algorithm on 6 large social, collaboration and information networks with *ground-truth* community information. Experiments show state of the art performance both in terms of the quality of detected communities as well as in speed and scalability of our algorithm.

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

1. INTRODUCTION

A large body of work in computer science, statistics, applied mathematics, and statistical physics has been devoted to identifying community structure in complex networks (see [8, 28, 32] for surveys of this area). A community (also referred to as a module or a cluster) is intuitively thought of as a group of nodes with more interactions amongst its members than between its members and the remainder of the network [10]. Such groups of nodes (*i.e.*, communities) are often interpreted as organizational units in social networks [7, 29], functional units in biochemical networks [17], ecological niches in food web networks [10], or scientific disciplines in citation and collaboration networks [3].



(a) No overlaps (b) Sparse overlaps (c) Dense overlaps

Figure 1: Three views on the structure of network communities. Present view of (a) two non-overlapping and (b) two overlapping communities. Present methods assume that the nodes in the overlap are less-well connected (b). We find densely connected community overlaps (c). Left: Network; Right: Corresponding adjacency matrix.

Even though methods for identifying overlapping as well as hierarchically-nested communities in networks have been considered in the past [1, 2, 25, 32], identifying meaningful communities in large networks has proven to be a challenging task [9, 20, 34]. Most methods have trouble scaling to large networks, and the lack of reliable "ground truth" makes evaluation of detected communities surprisingly difficult. Thus, while networks have been extensively-studied, and the existence and properties of communities in small networks is by now well-understood [1, 2, 10, 25], it is still not clear how to identify realistic overlapping communities in very large networks that are increasingly common.

Present work: Empirical observations. Our work starts with a novel, and in retrospective very intuitive, observation that overlaps of communities tend to be more densely connected than the non-overlapping parts [33, 35]. In particular, we empirically observe that the more communities a pair of nodes shares the more likely they are connected in the network. For example, people sharing multiple hobbies (*i.e.*, interest based communities) have a higher chance of becoming friends [23], researchers with many common interests (*i.e.*, many common scientific communities) are more likely to work and publish together [26].

Even though intuitive our observation is very subtle and represents a radical new view of networks communities and has important consequences for network community detection [33, 35]. To put our observation in the context, we first give a quick overview of recent developments in the network community detection. Traditionally, the emergence of communities in networks has been understood through the strength-of-weak-ties theory [12]. This theory led researchers to conceptualize networks as consisting of dense clusters that are linked by a small number of weak ties (Figure 1(a)). Graph partitioning [28, 30], modularity optimization [24]

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WSDM'13, February 4–8, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$15.00.

as well as betweenness centrality [10] based community detection methods all assume such view of network communities and thus aim to identify edges that can be cut in order to separate the network into a set of non-overlapping clusters.

In social as well as other types of networks nodes can belong to multiple communities simultaneously, which leads to overlapping community structure [25]. However, we noticed that practically all present overlapping community detection methods (for example [1, 2, 25]) make a hidden (and so far undocumented) assumption that community overlaps are *less densely* connected than the non-overlapping parts of communities (Figure 1(b)) [33, 35]. This leads to an unnatural modeling assumption that the *more* communities a pair of nodes shares, the *less* likely it is they are connected. Figure 1(b) illustrates the unnatural structure of community overlaps emerging under such assumption.

In contrast, we find an increasing relationship between the number of shared communities and the probability of nodes being connected by an edge [33, 35]. A direct consequence of this observation is that parts of the network where communities overlap tend to be *more densely* connected (Figure 1(c)) [35]. Even though very natural, the observation stands in sharp contrast to present definitions of network communities. More importantly, today's community detection methods (for example, [1, 2, 25]) cannot correctly identify such dense community overlaps. Present methods either mistakenly identify the overlap as a separate community or merge two overlapping communities into a single one [35].

Present work: Large-scale community detection via matrix factorization. Building on the above observation the goal of this work is to detect communities in a given large unlabeled undirected network. This means that, for every node in a given large undirected network, we aim to discover the communities it belongs to. To achieve this we develop a novel community detection method that allows for discovering any combination of densely overlapping, non-overlapping, as well as hierarchically nested communities.

We build on models of affiliation networks [5, 18] and develop the BIGCLAM (Cluster Affiliation Model for Big Networks). In BIGCLAM communities arise due to shared community affiliations of nodes. We explicitly model the affiliation strength of each node to each community. We assign each node-community pair a non-negative latent factor which represents the degree of membership of a node to the community. We then model the probability of an edge between a pair of nodes in the network as a function of the shared community affiliations.

We identify network communities by fitting the BIGCLAM model to a given large undirected network. Our goal is to estimate non-negative latent factors that model the membership strength of each node to each community. By combining the state-of-the-art non-negative matrix factorization methods [19] with block stochastic gradient descent [21] we achieve gains both in the quality of detected communities as well as in scalability of the method. We improve by a factor of 10 the size of the largest networks that overlapping community detection methods could process in the past.

An additional contribution of our work is improved evaluation. So far community detection methods have mostly been evaluated anecdotally on small networks. In contrast, we identify social, collaboration, information and biological networks with explicitly labeled ground-truth communities [34]. This allows for quantitative evaluating by assessing how well detected communities correspond to the ground-truth communities [34]. Experiments reveal that BIGCLAM discovers overlapping as well as non-overlapping community structure more accurately than present state-of-the-art methods [1, 2, 25, 27]. Moreover, BIGCLAM scales well beyond the current overlapping community detection methods. Experiments show that BIGCLAM achieves near linear running time while other methods exhibit quadratic or exponential running time. We process networks of more than 35 million edges which improves by a factor of 10 the size of the largest networks that overlapping community detection methods could process in the past. BIGCLAM improves over the current state of the art in both the scalability as well as the quality of detected communities. Code as well as all the data are available at `http://snap.stanford.edu`.

2. RELATED WORK

Our BIGCLAM is an example of a bipartite affiliation network model [5, 18, 36]. Affiliation networks have been extensively studied in sociology [5] as a metaphor of classical social theory concerning the intersection of persons with groups, where it has been recognized that communities arise due to shared group affiliations [5, 29]. In affiliation network models, nodes of the social network are affiliated with communities they belong to and the links of the underlying social network are then derived based on the node community affiliations. Whereas classical models assume binary node-community affiliations, in our model we also consider the strength of an affiliation which provides additional modeling flexibility.

BIGCLAM formulates community detection as a variant of non-negative matrix factorization (NMF) [15, 19, 21]. Similar to NMF, we aim to learn factors that can recover the adjacency matrix of a given network. However, BIGCLAM has two important improvements. First, most of NMF research pays relatively little attention to interpreting the latent factors. The primary goal there is to estimate the missing entries of the matrix (*e.g.*, as in the Netflix competition). On the other hand, BIGCLAM aims to learn latent factors which represent community affiliations of nodes. Second, instead of using a Gaussian distribution [15, 19] or logistic link function [14], we optimize the model likelihood of explaining the links of the observed network. Our formulation of likelihood allows us to compute a gradient of the factor matrix in near-constant time, which is significant improvement over existing NMF methods where the complexity of computing such gradient is *linear* in the number of rows of the matrix (*i.e.*, nodes of the network). In practice, computing the gradient in near-constant time makes our algorithm about 1,000 times faster.

In terms of scalability most overlapping community detection methods scale to networks with at most thousands of nodes [2, 22, 25]. The largest network processed with overlapping community detection methods is a mobile phone network of 800,000 nodes and 2.8 million edges [1]. Non-overlapping community detection algorithms, which solve a simpler problem, have been applied to networks with millions of nodes [6, 16]. Our methods presented here can process networks with tens of millions of edges while also obtaining state of the art quality of detected communities.

3. EMPIRICAL OBSERVATION

We motivate the development of our model by empirically studying the structure of communities and community overlaps in networks. We fist describe the network datasets with explicit ground-truth communities and then present our empirical findings.

Networks with ground-truth communities. To study the connectivity structure of community overlaps, we now describe networks with explicitly labeled ground-truth communities. To define such ground-truth, we collected 6 large social, information and collaboration networks where nodes explicitly state their community memberships [34]. Defining ground-truth communities will also help us later in evaluating the performance various methods (Section 6).

Dataset	N	E	C	S	A
LiveJournal	4.0 M	34.9 M	310 k	40.06	3.09
Friendster	120 M	2,600 M	1.5 M	26.72	0.33
Orkut	3.1 M	120 M	8.5 M	34.86	95.93
Youtube	1.1 M	3.0 M	30 k	9.75	0.26
DBLP	0.43 M	1.3 M	2.5 k	429.79	2.57
Amazon	0.34 M	0.93 M	49 k	99.86	14.83

Table 1: Dataset statistics. N: number of nodes, E: number of edges, C: number of communities, S: average community size, A: community memberships per node. M denotes a million and k denotes one thousand. On average 95% of all communities overlap with at least one other community.

First, we briefly describe the 6 networks [34][1]. The first 4 networks are online social networks: the LiveJournal blogging community, the Friendster online network, the Orkut social network, and the Youtube social network. Users in these networks create groups which other users then join. Such groups are formed over specific interests, hobbies, affiliations, and geographical regions. For instance, LiveJournal categorizes communities into the following types: culture, entertainment, expression, fandom, life/style, life/ support, gaming, sports, student life and technology. There are over 100 communities with 'Stanford' in their name, and they range from communities based around different classes, student ethnic communities, departments, activity and interest based groups, varsity teams, etc. We use such user-defined groups as ground-truth communities. A user can belong to zero, one or more ground-truth communities and thus ground-truth communities can overlap. The largest network among these online social networks is Friendster, which has 120 million nodes, 2.6 billion edges and 1.5 million ground-truth communities.

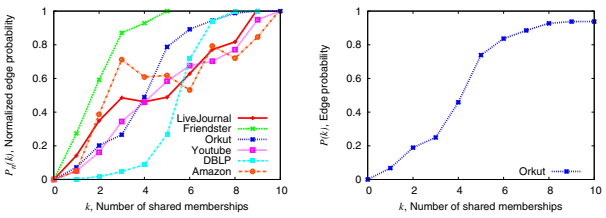
We also consider the Amazon product co-purchasing network where the nodes represent products and edges connect commonly co-purchased products. Each product (*i.e.*, node) belongs to one or more hierarchically nested product categories. We use each product category to define a ground-truth community. Members of the same community share a common function or a role. Ground-truth communities in the Amazon network can be overlapping or hierarchically nested. Last we also use the collaboration network of DBLP where nodes represent authors/actors and edges connect nodes that have co-authored a paper. Since research communities stem around conferences or journals, we use publication venues as ground-truth communities in DBLP.

The networks we consider show a nice range of scale in all measures (Table 1): The size of networks ranges from hundreds of thousands to hundreds of millions of nodes and edges and the number of ground-truth communities varies from hundreds to millions. Last, the networks represent a wide range of edge densities, numbers of explicit communities, as well as amounts of community overlap.

In our previous work [33, 34, 35] we found the above definitions of ground-truth to be reliable and robust. In particular, while the networks we consider here come from a variety of domains, span a wide range of network sizes and edge densities, we find our observations and results to be consistent and robust across all of them. The consistency and robustness of results make us confident in our methodology and empirical observations.

In order to express all networks in a consistent way we represent each network as an unweighted undirected static graph. Because members of the same group may be disconnected in the network, we treat each connected component of the group as a separate ground-truth community. We allow ground-truth communities to overlap because a node can belong to multiple groups at once.

[1]Networks are available at `http://snap.stanford.edu`.



(a) Edge probability (b) Orkut

Figure 2: (a) Normalized edge probability as a function of common memberships k. Probabilities are scaled so that maximum value over k is one. (b) Edge probability in the Orkut network, plotted as an absolute value. We conclude overlaps are more densely connected than single communities.

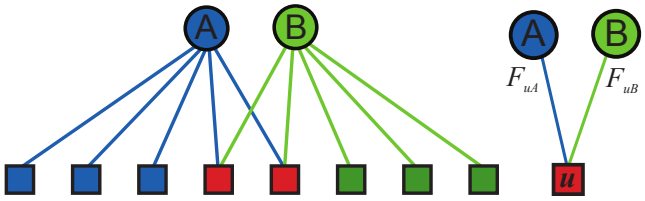
Observation: Community overlaps are dense. Having defined ground-truth communities, we now empirically study the structure of ground-truth communities. We find that ground-truth communities heavily overlap. On average 95% of all communities overlap with at least one other community and only 15% of community's members belong to only that community. We thus examine the structure of community overlaps by measuring the probability of a pair of nodes being connected given that they belong to k common communities, *i.e.*, the nodes reside in the overlap of same k communities. Figure 2(a) plots this probability for all six datasets. For visualization we scale each probability curve so that the maximum value of each curve over k is 1. Under the current assumption that overlaps are less dense than non-overlaps, the probability curves would decrease as k increases. In contrast, we notice an increasing relationship for all datasets, *i.e.*, the *more* communities a pair of nodes has in common, the *higher* the probability of an edge. This means that nodes residing in overlaps are more densely connected each other than the nodes in a single community [35].

To demonstrate how the edge probability changes as k increases, we plot the edge probability (without scaling) measured in the Orkut network as a function of the number of common communities k in Figure 2(b). Similar to all large networks, Orkut is extremely sparsely connected — the background probability of a random pair of nodes being connected is $\approx 10^{-5}$ — the increase in edge probability is highly significant. For example, if a pair of nodes has 2 communities in common, the probability of an edge is nearly 0.20. The edge probability increases by 10^4 times (from 10^{-5} to 10^{-1}) as soon as the pair share two communities.

Overall, in all the datasets we consistently observe similar and robust behavior: The probability of a pair of nodes being connected approaches 1 as the number of common communities increases [35]. While in online social networks the edge probability exhibits a diminishing-returns-like growth, in DBLP, it appears to follow a threshold-like behavior.

Discussion. In retrospective, the above observation is very intuitive and thus so much more surprising. For pairs of nodes that belong to multiple common communities, edges often exist due to one dominant reason. Thus, nodes in the overlaps will have higher chance of being connected because they belong to multiple communities. Many examples to support this. For example, people sharing multiple hobbies or belonging to several common institutions have a higher chance of becoming friends [23], researchers with many common interests are more likely to work together [26], and proteins involved in multiple common functional modules are more likely to interact [17].

The observation that the probability of an edge *increases* as a function of the number of shared communities means that nodes in the overlap of two (or more) communities are more likely to be



(a) Community Affiliation Network (b) Weight for affiliation

Figure 3: (a) Bipartite community affiliation graph. Circles: Communities, Squares: Nodes of the underlying network. Edges indicate node community memberships. Edges with zero weight are not shown. (b) Each affiliation edge from node u to community c has strength $F_{uc} \geq 0$.

connected. Thus, our finding suggests communities overlap as illustrated in Figure 1(c) where the overlap of the two communities is more densely connected than each single community. However, we note that our finding is in sharp contrast to the currently predominant view of network communities which is based on two fundamental social network theories: triadic closure [31] and strength-of-weak-ties [12]. This leads to the picture of network communities as illustrated in Figure 1(a) which suggests that homophily in networks operates in small pockets where nodes gather in dense non-overlapping clusters. Extending these two theories to the overlapping communities leads to the unnatural structure of community overlaps as illustrated in Figure 1(b): Community overlaps are *less densely* connected than the groups themselves. Our results show the contrary. As a consequence this means that present overlapping community detection methods [1, 2, 25] which rely on the assumption of sparse overlaps fail to correctly identify dense community overlaps. They would either merge two overlapping communities into a single cluster or identify the overlap as a separate cluster [35].

Last, we also note that the observation that community overlaps are denser than communities themselves nicely extends the notion of homophily in networks [23]. The 'strength of weak ties' [12] suggests that homophily in networks operates in small pockets where inside the pocket nodes link strongly among themselves, and weakly to other pockets. Our work extends the understanding of homophily. We are discovering *pluralistic homophily* where the similarity of one node to another is the number of shared affiliations, not just their similarity along a single dimension. This view of tie formation is consistent with the works of Simmel [29] on the web of affiliations, and Feld [7] on focused organization of social ties. In both of these views networks consist of overlapping "tiles" or "social circles" that serve as organizing principles of nodes in networks.

4. CLUSTER AFFILIATION MODEL

Next we present the *Cluster Affiliation Model for Big Networks* (BIGCLAM), a probabilistic generative model for graphs that reliably captures the organization of networks based on community affiliations. Our model has three main ingredients:

The first ingredient is based on Breiger's work [5] which recognized that communities arise due to shared group affiliations [5, 7, 29]. We represent node community memberships with a bipartite affiliation network that links nodes of the social network to communities that they belong to (Figure 3(a)).

The second ingredient stems from the fact that people tend to be involved in communities to various degrees. Therefore, we assume that each affiliation edge in the bipartite affiliation network has a nonnegative weight. The higher the node's weight of the affiliation to the community the more likely is the node to be connected to other members in the community.

The last ingredient of our model is based on the fact that, when people share multiple community affiliations (*e.g.*, co-workers who attended the same university), the links between them stem for one dominant reason (*i.e.*, shared community). This means that for each community a pair of nodes shares we get an independent chance of connecting the nodes. Thus, naturally, the more communities a pair of nodes shares, the higher the probability of being connected.

Figure 3 illustrates our model. We start with a bipartite graph where the nodes at the bottom represent the nodes of the social network G, the nodes on the top represent communities C, and the edges M indicate node community affiliations. We denote the bipartite affiliation network as $B(V, C, M)$.

The flexibility of the affiliation network allows us to model a wide range of network community structures. Figure 4 illustrates the structure of the network as well as the corresponding node-community affiliation network. Figure 4(a) shows an affiliation graph of a network with two non-overlapping communities. The affiliation graph in Figure 4(c) represents hierarchical community structure where communities A and C are nested inside community B. Finally, Figure 4(b) shows an example of overlapping communities. These three very different examples demonstrate that the flexibility of the affiliation network structure allows BIGCLAM to simultaneously model any combination of non-overlapping, hierarchically nested as well as overlapping communities in networks.

From node-community affiliations to the edges of the network. To generate a network $G(V, E)$ given a bipartite community affiliation $B(V, C, M)$ we need to specify the process that generates the edges E of G given the affiliation network B. We consider a simple parameterization where we assign a nonnegative weight F_{uc} between node $u \in V$ and community $c \in C$. ($F_{uc} = 0$ means no affiliation.) Given F, we assume that each community c connects its member nodes depending on the value of F. In particular, each community c connects its member nodes u, v with probability $1 - \exp(-F_{uc} \cdot F_{vc})$. Each community c creates edges independently. However, if a pair of nodes gets connected multiple times, the duplicate edges are not included in the graph $G(V, E)$. Since each community c connects u, v independently with probability $1 - \exp(-F_{uc} \cdot F_{vc})$, the edge probability between u and v is $1 - \exp(-\sum_c F_{uc} \cdot F_{vc})$ and thus increasing in the number of shared communities.

DEFINITION 1. *Let F be a nonnegative matrix where F_{uc} is a weight between node $u \in V$ and community $c \in C$. Given F, the* BIGCLAM *generates a graph $G(V, E)$ by creating edge (u, v) between a pair of nodes $u, v \in V$ with probability $p(u, v)$:*

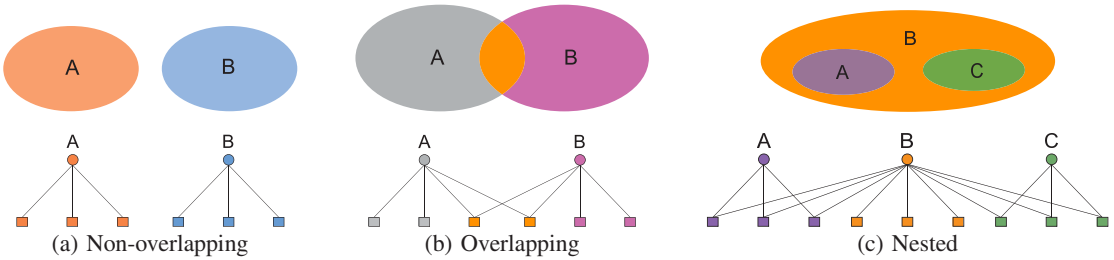
$$p(u, v) = 1 - \exp(-F_u \cdot F_v^T), \tag{1}$$

where F_u is a weight vector for node u ($F_u = F_{u\cdot}$).

The process in Eq. 1 suggests the following probabilistic interpretation. Assume an undirected weighted network where pairs of nodes have a latent interaction of non-negative strength X_{uv}. However, we only observe an undirected unweighted version of network $G(V, E)$ where a pair of nodes u, v is connected if the corresponding $X_{uv} > 0$. Now consider that nodes u, v generate an interaction of strength $X_{uv}^{(c)}$ within each community c using a Poisson distribution with mean $F_{uc} \cdot F_{vc}$. Then the total amount of interaction X_{uv} between nodes u and v is the sum of $X_{uv}^{(c)}$:

$$X_{uv} = \sum_c X_{uv}^{(c)}, \quad X_{uv}^{(c)} \sim Pois(F_{uc} \cdot F_{vc}).$$

Then, due to the additivity of the Poisson random variable, $X_{uv} \sim Pois(\sum_c F_{uc} \cdot F_{vc})$, and the edge probability ($P(X_{uv} > 0)$) is the same as $p(u, v)$ in Eq. 1.



(a) Non-overlapping (b) Overlapping (c) Nested

Figure 4: BIGCLAM allows for rich modeling of network communities: (a) non-overlapping, (b) overlapping, (c) nested. In (a) we assume that nodes in two communities connect with small prob. ε (refer to the discussion in the main text).

$$P(X_{uv} > 0) = 1 - P(X_{uv} = 0) = 1 - \exp(-\sum_c F_{uc} \cdot F_{vc})$$

Note that node u with higher F_{uc} is more likely to be connected to other members of c as $X_{uv}^{(c)}$ will have a higher mean.

Note that this process naturally generates an increasing relationship between edge probability and the number of shared communities. This is due to the fact that nodes that share multiple community memberships receive multiple chances to create a link. For example, pairs of purple nodes in the overlap of communities A and B in Figure 3(a) get two chances to create an edge. First they create an edge with probability $1 - e^{-F_{uA} \cdot F_{vA}}$ (due to the membership to community A) and then also an edge with probability $1 - e^{-F_{uB} \cdot F_{vB}}$ (due to membership to community B). The edge probability between these nodes is $1 - e^{-(F_{uA} \cdot F_{vA} + F_{uB} \cdot F_{vB})}$. If they were to reside in the non-overlapping region of A, they would be linked with probability $1 - e^{-F_{uA} \cdot F_{vA}}$, which is smaller than $1 - e^{-(F_{uA} \cdot F_{vA} + F_{uB} \cdot F_{vB})}$.

ε-Community. In the formulation of Equation 1, BIGCLAM does not allow for the edges between the nodes u and v that do not share any common communities since for such nodes $F_{uc} \cdot F_{vc} = 0$ for all c. To allow for edges between nodes that do not share any community affiliations, we assume an additional community, called the ε-community, which connects *any* pair of nodes with a very small probability ε. We find that setting ε to be the background edge probability between a random pair of nodes ($\varepsilon = 2|E|/|V|(|V|-1)$) works well in practice. For all our experiments we set $\varepsilon \approx 10^{-8}$.

5. COMMUNITY DETECTION

Now that we defined the BIGCLAM model, we explain how to detect network communities using the model. Given an unlabeled undirected network $G(V, E)$, we aim to detect K communities by *fitting* the BIGCLAM (*i.e.*, finding the most likely affiliation factor matrix $\hat{F} \in \mathbb{R}^{N \times K}$) to the underlying network G by maximizing the likelihood $l(F) = \log P(G|F)$ of the underlying G:

$$\hat{F} = \underset{F \geq 0}{\operatorname{argmax}}\, l(F), \tag{2}$$

where

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

For now, we assume the number of communities K is given. We will describe later how to automatically estimate K.

The optimization problem of Eq. 2 can be viewed as a variant of nonnegative matrix factorization (NMF) [19] where we learn $F \in \mathbb{R}^{N \times K}$ that best approximates the adjacency matrix A of a given network G. By representing a negative log-likelihood $-l(F)$ as a loss function D and $1 - exp(\cdot)$ as a link function, we can represent the problem as follows:

$$\hat{F} = \underset{F \geq 0}{\operatorname{argmin}}\, D(A, f(FF^T))$$

The benefit of using matrix factorization approach is increased scalability. Overlapping community detection methods have been developed to analyze small networks [1, 2], and most methods rely on combinatorial optimization which is hard to scale. On the other hand, for nonnegative matrix factorization many efficient techniques exist [15, 21].

BIGCLAM modifies the existing NMF methods [15, 19, 21] and adapts them to large networks. While NMF methods use l_2 norm as an objective function, l_2 norm is not suitable for modeling binary adjacency matrices [14]. Instead, BIGCLAM employs log-likelihood as a loss function. Additional benefit is that for sparsely connected networks (which real networks are) our formulation allows for near-constant time gradient computation (l_2 takes linear time) which in practice speeds up our algorithm for a factor of 100.

Solving the optimization problem. To solve the problem in Eq. 2, we adopt a block coordinate gradient ascent algorithm [15, 21]. In particular, we update F_u for each u with the other F_v fixed, *i.e.*, we update the memberships of one node with fixing the membership of all other nodes. The main reason is that if we fix all F_v, then the problem of updating F_u becomes a convex optimization problem. We solve the following subproblem for each u:

$$\underset{F_{uc} \geq 0}{\operatorname{argmax}}\, l(F_u), \tag{3}$$

where

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

where $\mathcal{N}(u)$ is a set of neighbors of u. To solve this convex problem, we use projected gradient ascent. The gradient can be computed straightforwardly.

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

We compute a step size using backtracking line search [4]. After update, we project F_u into a space of nonnegative vectors by setting $F_{uc} = \max(F_{uc}, 0)$.

For a large network with more than a million nodes, this coordinate ascent is not very scalable as making a single step of coordinate ascent (*i.e.*, computing $l(F_u)$ and $\nabla l(F_u)$) takes linear time $O(N)$. However, we reduce the complexity to $O(|\mathcal{N}(u)|)$ by computing $\sum_{v \notin \mathcal{N}(u)} F_v$ efficiently. In particular, we notice:

$$\sum_{v \notin \mathcal{N}(u)} F_v = (\sum_v F_v - F_u - \sum_{v \in \mathcal{N}(u)} F_v) \tag{4}$$

By storing $\sum_v F_v$, we can compute $\sum_{v \notin \mathcal{N}(u)} F_u F_v^T$ in time $O(|\mathcal{N}(u)|)$. 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. We iteratively update F_u for each u and stop the iteration if the likelihood does not increase (increase less than 0.001%) after we update F_u for all u. In practice this speeds up our algorithm for two orders of magnitude and makes it practical to run it on networks with millions of nodes and edges.

Determining community affiliations. After we learn $\hat{F}$, we still have to determine whether u belongs to community c or not from the value of F_{uc}. To achieve this, we ignore the membership of node u to community c if F_{uc} is below some threshold δ. Otherwise ($F_{uc} \geq \delta$), we regard u as belonging to c. We set δ so that if two nodes belong to community c, then their edge probability is higher than the background edge probability ε (see Section 4).

$$\varepsilon \leq 1 - \exp(-\delta^2)$$

Solving this inequality, we set the value of $\delta = \sqrt{-\log(1-\varepsilon)}$. Note we also experimented with other values of δ and found that our choice for δ gives overall good performance.

Initialization. To initialize F, we use locally minimal neighborhoods [11]. Neighborhoods $N(u)$ of node u is a community of u and its neighbors, and $N(u)$ is locally minimal if $N(u)$ has lower conductance than all the $N(v)$ for nodes v who are connected to u. Recently, Gleich et al. [11] empirically showed that the locally minimal neighborhoods are good seed sets for community detection algorithms. For a node u' who belongs to a locally minimal neighborhood k, we initialize $F_{u'k} = 1$, otherwise $F_{u'k} = 0$.

Choosing the number of communities. To find the number of communities K, we adopt the approach used in [2]. We reserve 20% of node pairs as a hold out set. Varying K, we fit the BIGCLAM model with K communities on the 80% of node pairs and then evaluate the likelihood of BIGCLAM on the hold out set. The K with the maximum hold out likelihood will be chosen as the number of communities. When the network is too small (*e.g.*, has less than 50 edges), we use K that achieves the smallest value of the Bayes Information Criterion:

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

Implementational details. Since the objective function of our optimization problem is not the l_2 norm, the methods for least squares NMF such as multiplicative update [19] or alternating least squares [15] are not applicable. We experimented with the cyclic coordinate descent method (CCD) [15] which optimizes F_{uc} for each u and each c by the Newton's method, but the method converged slower than our block coordinate ascent method. The main reason for this is that the number of subproblems that we have to solve in CCD grows linearly with K, the number of communities. In matrix factorization, usually K (the rank of F) is assumed to be a very small constant [15, 21]; however, in our problem K increases as the size of the underlying network grows.

Connection to other affiliation network models. Last we also briefly describe the connection between BIGCLAM and other affiliation network models. In particular, we consider the AGM [35, 33] which can also model densely overlapping network community structure. Similarly to BIGCLAM, AGM generates $G(V, E)$ given a bipartite community affiliation $B(V, C, M)$. In contrast to BIGCLAM, AGM assigns a single parameter p_c to every community c. Given $B(V, C, M)$ and $\{p_c\}$, AGM models the edge probability $p(u, v)$ as follows:

$$p(u, v) = 1 - \prod_{c \in C_{uv}} (1 - p_c).$$

where C_{uv} is a set of communities that u and v have in common.

One can also detect community structure by fitting AGM to a given network $G(V, E)$ (*i.e.*, finding affiliation graph B and parameters $\{p_c\}$) by maximizing the log-likelihood [33]:

$$\underset{P,\{p_c\}}{\operatorname{argmax}} \sum_{(u,v)\in E} \log p(u, v) + \sum_{(u,v)\notin E} \log(1 - p(u, v)) \qquad (5)$$

This results in a combinatorial optimization problem that is very hard to solve. Solving the problem requires a combinatorial search over all possible affiliation graphs B. However, there is an exponential number ($2^{N \cdot K}$) of possible affiliation graphs B.

We now show that fitting BIGCLAM (Eq. 2) can also be derived by relaxing the fitting problem of AGM (Eq. 5) into a continuous optimization problem. We begin by stating Eq. 1 in a new form:

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

where M_{uc} is an indicator variable whether node u belongs to community c. By replacing $1 - p_c = \exp(-\alpha_c)$ with $\alpha_c \geq 0$, we can express the equation as a linear form of M and α_c:

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

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

$$p(u, v) = 1 - \exp(-\tilde{M}_u \tilde{M}_v^T).$$

Note that we did not use any approximation so far. So the maximum likelihood estimation of the model is still a combinatorial optimization problem ($\tilde{M}_{uc} \in \{\sqrt{\alpha_c}, 0\}$). $\tilde{M}_{uc} \in \{\sqrt{\alpha_c}, 0\}$ means that if node u belongs to c, it would be connected to other member nodes in c with the factor $\sqrt{\alpha_c}$. Therefore, we can interpret $\tilde{M}_{uc}$ as the level of participation of u in community c, which then determines edge probability of u to other nodes in c. Basically, we can replace $\tilde{M}_{uc}$ with a *continuous* membership F_{uc} which can be any nonnegative number. This way we actually model a level of participation of each node in a particular community as members with the higher value of F_{uc} will be more likely to connect to other members of c.

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

Now, we transform the problem of Eq. 5 into a continuous optimization problem:

$$\hat{F} = \underset{F \geq 0}{\operatorname{argmax}} \sum_{(u,v)\in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v)\notin E} F_u F_v^T.$$

In other words, we can view the optimization problem of BIGCLAM as a continuous relaxation of the combinatorial optimization problem of fitting AGM. BIGCLAM can be considered as a relaxed version of AGM in the sense that it models community affiliation as continuous variables. With BIGCLAM, finding the most probable community affiliation is equivalent to factorizing the adjacency matrix of the underlying network with nonnegative factors.

6. EXPERIMENTS

We proceed by evaluating the performance of BIGCLAM and comparing it to the state-of-the-art community detection methods on a range of networks from a number of different domains and research areas.

6.1 Experiments on synthetic networks

Using synthetic networks we investigate the scalability and convergence of the BIGCLAM optimization problem.

Convergence of BIGCLAM. Non-negative matrix factorization is non-convex which means that gradient based approaches do not guarantee to find an optimal solution. To verify that our fitting algorithm does not suffer too much from local optima, we conduct the following experiment on synthetic networks. We generated 100 synthetic networks using the AGM model [35]. For each of these networks, we then fit BIGCLAM using 10 different random starting points and attempt to recover the true community affiliations.

In 98% of cases our fitting algorithm finds true communities with reliable accuracy (F1-score of node community memberships higher than 0.85), and in 27% of cases our algorithm discovers the communities almost perfectly (F1-Score > 0.95). This result suggests that the optimization space has several local optima which almost equivalent to the global optimum.

Scalability of BIGCLAM. We also evaluate the scalability of BIGCLAM by measuring the running time on the networks of increasing sizes.For comparison, we compare the runtime of the following overlapping community detection methods:

- NMF: Least squares non-negative matrix factorization. We solve the following problem: $\text{argmax}_{F_{uk} \geq 0} ||A - F \cdot F^T||_F$ where A is an adjacency matrix of a given network. We used a projected gradient descent as we do with BIGCLAM.
- BIGCLAM(Naive): BIGCLAM without the optimization in Eq. 4.
- LC: Link Clustering method [1].
- CPM: Clique Percolation method [25].
- MMSB: Mixed-Membership Stochastic Blockmodel [2].

Link Clustering, Clique Percolation Method and Mixed Membership Stochastic Blockmodels are considered the state-of-the-art overlapping community detection methods. We used the implementation of LC and CPM in the Stanford Network Analysis Platform[2]. For MMSB we used publicly-available 'LDA' R package. For CPM, we use the clique size $k = 5$ for CPM. For MMSB, we set the number of communities to detect to $K = 10$. We also consider NMF and BIGCLAM (Naive) so that we can compare the performance gain due to the optimization described in Eq. 4.

Figure 5 shows the results. NMF, BIGCLAM(Naive) and MMSB scale to networks of around 1,000. LC and CPM scale to networks of about 10,000 and then their runtime becomes prohibitively large. On the other hand BIGCLAM can process networks with hundreds of thousands of nodes within 20 minutes. This means that BIGCLAM can easily process networks 10 to 100 times larger than other approaches (and while also more accurately detecting communities). Last, note that the optimization of BIGCLAM defined in Eq. 4 speeds up the algorithm for around 100 times and is thus essential for making BIGCLAM scale to large networks.

6.2 Experiments using real ground-truth

We also examine the performance of BIGCLAM using the 6 networks with ground-truth communities that we described in Section 3. In these networks nodes explicitly state their ground-truth community memberships which allows us to quantify the 'accuracy' of community detection methods by evaluating the level of correspondence between detected and ground-truth communities.

Experimental setup. We are given an unlabeled undirected network G (with known ground-truth communities C^*) we aim to discover communities $\hat{C}$ such that discovered communities $\hat{C}$ closely match the ground-truth communities C^*.

[2]SNAP: `http://snap.stanford.edu/snap`

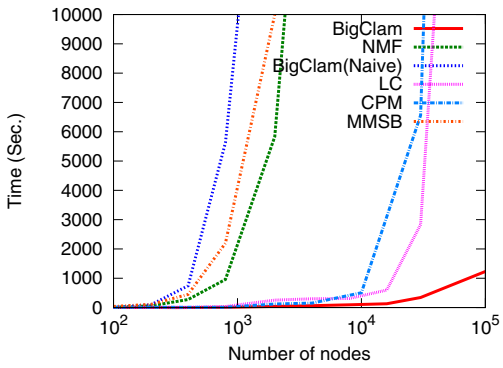


Figure 5: Algorithm runtime comparison. BIGCLAM runs 10 to 100 faster than competing approaches.

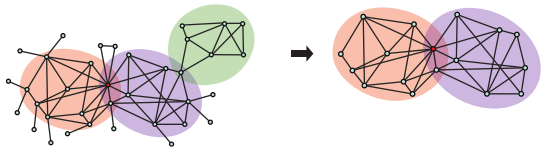


Figure 6: Sampling subnetworks of G.

Even though our algorithm can process the networks described in Table 1, *all* the baseline methods do not scale to networks of such size. To allow for comparison between our and the baseline methods we use the following evaluation scenario where the goal is to obtain a large set of relatively small subnetworks with overlapping community structure. To obtain one such subnetwork we pick a random node u in the given graph G that belongs to at least two communities. We then take the subnetwork to be the induced subgraph of G consisting of all the nodes that share at least one ground-truth community membership with u. Figure 6 illustrates how a subnetwork (right) is created from $G(V, E)$ (left) based on the red node u. Note that on average 95% of all ground-truth communities overlap which means that this procedure does not bias towards overlapping communities. In our experiments we created 500 different subnetworks for each of the six datasets.

Baselines for comparison. For baselines we choose three most prominent overlapping community detection methods: Link clustering (LC) [1], Clique Percolation Method (CPM) [25], and the Mixed-Membership Stochastic Block Model (MMSB) [2].

These methods have a number of parameters that need to be set. For CPM, we set the clique size $k = 5$ since the number of communities discovered by CPM with $k = 5$ best approximates the true number of communities. For MMSB, we have to set the number of communities K as an input parameter. We use the Bayes Information Criterion to choose K. While we require "hard" community memberships, MMSB returns stochastic node memberships to each of the K communities. Thus, we assign a node to a community if the corresponding stochastic membership is non-zero. We also considered Infomap [27], which is the-state-of-the-art non-overlapping community detection method. We omit the results as the performance of the method was not competitive.

Evaluation metrics. The availability of ground-truth communities allows us to *quantitatively* evalute the performance of community detection algorithm. Without ground-truth such evaluation is simply not possible. For evaluation we use metrics that quantify the level of correspondence between the detected and the ground-truth communities. Given a network $G(V, E)$, we consider a set of

ground truth communities C^* and a set of detected communities $\hat{C}$ where each ground-truth community $C_i \in C^*$ and each detected community $\hat{C_i} \in \hat{C}$ is defined by a set of its member nodes. To quantify the level of correspondence of $\hat{C}$ to C^* we consider:

- **Average F1 score.** To compute the F1 score, we need to determine which $C_i \in C^*$ corresponds to which $\hat{C}_i \in \hat{C}$. We define F1 score to be the average of the F1-score of the best-matching ground-truth community to each detected community, and the F1-score of the best-matching detected community to each ground-truth community:

$$\frac{1}{2}\left(\frac{1}{|C^*|}\sum_{C_i \in C^*} F1(C_i, \hat{C_{g(i)}}) + \frac{1}{|\hat{C}|}\sum_{\hat{C_i} \in \hat{C}} F1(C_{g'(i)}, \hat{C_i})\right)$$

  where the best matching g and g' is defined as follows:

$$g(i) = \operatorname*{argmax}_j F1(C_i, \hat{C_j}), \quad g'(i) = \operatorname*{argmax}_j F1(C_j, \hat{C_i})$$

  and $F1(C_i, \hat{C_j})$ is the harmonic mean of Precision and Recall.

- **Omega Index [13]** is the accuracy on estimating the number of communities that each pair of nodes shares:

$$\frac{1}{|V|^2}\sum_{u,v \in V} \mathbf{1}\{|C_{uv}| = |\hat{C_{uv}}|\}$$

  where C_{uv} is the set of ground-truth communities that u and v share and $\hat{C_{uv}}$ is the set of detected communities that they share.

- **Normalized Mutual Information** adopts the criterion used in information theory to compare the detected communities and the ground-truth communities. Normalized Mutual Information has been proposed as a performance metric for community detection. Refer to [8] for details.

- **Accuracy in the number of communities** is the relative accuracy between the detected and the true number of communities, $1 - \frac{||C^*| - |\hat{C}||}{2|C^*|}$.

For all metrics higher values mean more "accurately" detected communities, *i.e.*detected node community memberships better correspond to ground-truth node community memberships. Maximum value of 1 is obtained when the detected communities perfectly correspond to the ground-truth communities.

Results on ground-truth communities. For each community detection method and each dataset we measure the average value of the 4 evaluation metrics over the 500 subnetworks sampled using the procedure described above. Then, for each evaluation metric separately we scale the scores of the methods so that the best performing community detection method achieves the score of 1. Finally, we compute the composite performance by summing up the 4 normalized scores. If a method outperforms all the other method in all the scores, then its composite performance is 4.

Figure 7 displays the composite performance of the methods over all six networks. On average, the composite performance of BIGCLAM is 3.60, which is 79% higher than that of Link clustering (2.01), 45% higher than that of CPM (2.47), and 15% higher than that of MMSB (3.14). The absolute average value of Omega Index of BIGCLAM over the 6 networks is 0.47, which is 24% higher than Link clustering (0.38), 26% higher than CPM (0.37), and 30% higher than MMSB (0.36). In terms of absolute values of scores, BIGCLAM archives the average F1 score of 0.60, average Omega index of 0.47, Mutual Information of 0.22 and accuracy of the number of communities of 0.43.

Overall, BIGCLAM gives superior overall performance. This means that, while BIGCLAM is two orders of magnitude more scalable than competing approaches, it also achieves superior performance in the quality of detected communities. On 4 out of 6 networks BIGCLAM performs best by a big margin. However, we note that on DBLP and Amazon MMSB is the winning method mostly due to BIGCLAM scoring very badly on a single individual metric (Number of communities on DBLP, Ω-index on Amazon). This occurs due to the fact that BIGCLAM uses a single parameter ε to model the edge probability between all pairs of different communities (ε-Community in Section 4), while MMSB uses one parameter for each pair of communities. With more parameters, MMSB can fit these networks better. Note that BIGCLAM could be easily extended to include a distinct parameter for the edge probability between each pair of communities.

6.3 Experiments on networks in Ahn et al. [1]

We further evaluate BIGCLAM using performance benchmarks from Ahn et al. [1]. For this experiment we adopt exactly the same data, evaluation metrics and experimental setup as in [1]. Note that these networks do not contain information about ground-truth communities. However, nodes in these communities contain attributes and [1] used "purity" metrics as surrogates for the quality of detected communities. The idea behind evaluation metrics here is that good communities have low diversity of member nodes' features.

Experimental setup. We use the same seven different networks as in [1]: 5 biological networks, a network of Wikipedia pages and a word association network For further details about these datasets, refer to [1]. We also adopt the same data-driven measures defined in [1]: Community Coverage, Overlap Coverage, Community Quality, Overlap Quality. All networks are small, so we apply the community detection methods to full networks. Moreover, the metrics are heavily biased towards methods that find a large number of communities, so we fit BIGCLAM using the same number of communities as detected by LC (*i.e.*, the algorithm developed in [1]).

Results. Following [1] we compute the composite performance by normalizing the scores the same way as we did in the experiments with ground-truth communities. Figure 8 shows the composite performance of the four methods. The BIGCLAM achieves best composite performance in 4 networks, and the second best in three networks. In all these cases MMSB slightly outperforms BIGCLAM due to BIGCLAM's bad performance on the Overlap Coverage metric.Overlap Coverage is defined as the average number of communities that a node belongs to [1]. This metric is extremely ill posed since assigning nodes to more communities always improves the score. Since any non-zero stochastic membership found by MMSB is regarded as a valid community membership, the MMSB achieves extremely high score on the Overlap Coverage metric. Nevertheless, on average, the BIGCLAM achieves a composite performance score of 3.06, outperforming Link clustering (2.67) by 14%, Clique percolation (1.50) by 102%, and MMSB (2.84) by 8%.

6.4 Experiments on large networks

In addition to better accuracy, another strength of BIGCLAM is its scalability. To test this, we apply BIGCLAM to large real-world networks. We were able to run BIGCLAM on 4 (full) networks from Table 1: LiveJournal, Youtube, Amazon, and DBLP.

To reduce the memory requirements of our method, we aim to find sparse latent factors. We achieve this by adding l_1 regularization term to Eq. 2 and optimize:

$$\operatorname*{argmax}_{F_{uc} \geq 0} l(F) - \lambda \sum_{u,c} |F_{uc}|$$

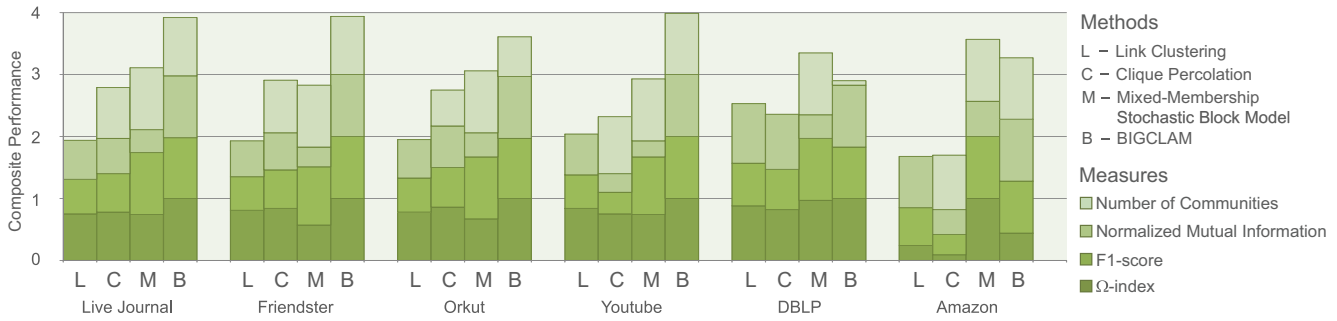
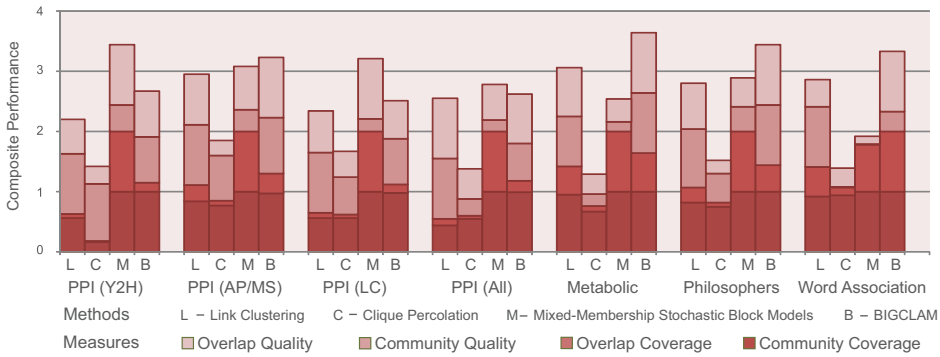


Figure 7: Performance of detecting ground-truth communities. While being 10 to 100 times faster than competing approaches BIGCLAM also achieves overall best performance in the "accuracy" of detected communities.



Network	N	E
PPI (Y2H)	1,647	2,518
PPI (AP/MS)	1,004	8,319
PPI (LC)	1,213	2,556
PPI (All)	1,647	12,784
Metabolic	1,042	8,756
Philosophers	1,218	5,972
Word Association	5,018	55,232

Figure 8: Experiments on the data and evaluation metrics used in Ahn et al. [1]. N: Number of the nodes, E: Number of the edges.

Since l_1 regularization introduces sparsity to matrix F, we only need to keep track of latent factors with non-zero value, which decreases the memory requirements of our method. We use $\lambda = 10$ for Amazon, Youtube, and DBLP and $\lambda = 5000$ for LiveJournal. We update F_u (Solving Eq. 3) for multiple nodes in parallel. With 20 threads, it takes about one day to fit BIGCLAM to the LiveJournal network (4M nodes, 35M edges).

As our baselines from the previous experiments do not scale to these networks, we consider two well-known graph partitioning methods as baselines: Metis [16] and Graclus [6]. For Graclus and Metis, we set the number of communities to detect K to be the number of ground-truth communities and use the same K for BIGCLAM as well.

Similarly to experiments in Figure 7, we measure the accuracy of detected communities using F-1 score and Omega index (NMI is omitted as all the methods perform the same). Moreover, notice that grund-truth communities in our data are partially annotated as some nodes might not indicate their memberships.This means it is important to quantify the Recall of a given method. We define Recall as the average Recall of best-matching detected communities:

$$Recall(C^*, \hat{C}) = \frac{1}{|C^*|} \sum_{C_i \in C^*} Rc(C_i, C^{\hat{}}_{g(i)})$$

where $\text{Rc}(C_i, \hat{C}_j)$ is the recall of $\hat{C}_j$ under the best matching g.

Since the two baselines (Graculus and Metis) perform very similarly in all metrics, we take just the best value among the two in each case rather than showing the result of baselines separately. For each network and each score, we pick the best score x among the two baselines and compute the relative improvement of BIGCLAM over the x, *i.e.*, $\frac{Score(\text{BIGCLAM}) - x}{x}$. Table 2 shows the relative improvement of BIGCLAM over the baselines. For example, 0.21 for F-1 in LiveJournal means that BIGCLAM achieves 21% higher F-1 score than the best baseline (Metis in this case).

Dataset	Ω-Index	F-1	Recall
LiveJournal	2.70	0.21	0.43
Youtube	1.60	0.39	0.82
Amazon	0.00	0.00	0.23
DBLP	0.10	0.03	0.29
Average	1.10	0.16	0.44

Table 2: Relative improvement of BIGCLAM over Metis and Graclus in detecting communities in large scale networks. Positive value indicates that BIGCLAM outperforms the baselines.

Overall, BIGCLAM outperforms the baselines in nearly all cases. On average, BIGCLAM achieves 110% higher Omega index, 16% higher F-1 score, and 44% higher average Recall, which means that BIGCLAM achieves 57% relative improvement on average among the three scores. Furthermore, BIGCLAM outperforms the baselines in every measure and every network. The absolute value of the scores of BIGCLAM is 0.11 (Omega index), 0.13 (F-1 score), and 0.32 (Recall). Overall, the results emphasize the need for a scalable and accurate overlapping community detection method as graph partitioning methods fail to detect overlapping communities. Results demonstrate that BIGCLAM could be the needed solution.

7. CONCLUSION

In this paper we developed a novel large scale community detection method that accurately discovers the overlapping community structure of real-world networks. We identified a set of networks where nodes explicitly state their ground-truth community membership and studied the connectivity of ground-truth communities and their overlaps. We observed that the overlaps of communities are more densely connected than the non-overlapping parts of

communities, which is in sharp contrast to assumptions made by present community detection models and methods. Based on this observation, we then developed the *Cluster Affiliation Model for Big Networks* (BIGCLAM), a conceptual model of network community structure, which naturally produces dense community overlaps. We then presented an efficient algorithm to fit BIGCLAM to a given network. Our fitting algorithm builds on the research of nonnegative matrix factorization and scalable to networks with million nodes. Experiments show that the BIGCLAM outperforms the state-of-the-art community detection methods in accurately discovering network communities as well as the overlaps between communities. Furthermore, BIGCLAM can detect community structure in the LiveJournal network which is more than 10 times bigger than the previously largest network considered for overlapping community detection.

Our work has several implications: First, our analysis sheds light on the organization of complex networks and provides new directions for research on community detection. Second, ground-truth communities offer a reliable way for evaluating community detection methods. Third, large scale overlapping community detection by BIGCLAM can broaden our understanding of organizing principles of large scale networks. And last, BIGCLAM opens up a new possibility to combine the advances in community detection and nonnegative matrix factorization. More generally, a shift in perspective from sparse to dense community overlaps represents a new way of studying networks and provides a unifying framework for network community detection.

Acknowledgements. This research has been supported in part by NSF IIS-1016909, CNS-1010921, IIS-1149837, IIS-1159679, DARPA XDATA, DARPA GRAPHS, Albert Yu & Mary Bechmann Foundation, Allyes, Boeing, Docomo, Intel, Samsung, Alfred P. Sloan Fellowship and the Microsoft Faculty Fellowship.

8. REFERENCES

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

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

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06*, 2006.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] R. L. Breiger. The duality of persons and groups. *Social Forces*, 53(2):181–190, 1974.

[6] I. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE PAMI*, 29(11):1944–1957, 2007.

[7] S. L. Feld. The focused organization of social ties. *American J. of Sociology*, 86(5):1015–1035, 1981.

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

[9] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *PNAS*, 104(1):36–41, 2007.

[10] M. Girvan and M. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

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

[12] M. S. Granovetter. The strength of weak ties. *American J. of Sociology*, 78:1360–1380, 1973.

[13] S. Gregory. Fuzzy overlapping communities in networks. *J. of Stat. Mech.: Theory and Experiment*, 2011.

[14] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon. Low-rank modeling of signed networks. In *KDD '12*, 2012.

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

[16] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48:96–129, 1998.

[17] N. Krogan et al. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

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

[19] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.

[20] 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*, 6(1):29–123, 2009.

[21] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 2007.

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

[23] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[24] M. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[25] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[26] W. W. Powell, D. R. White, K. W. Koput, and J. Owen-Smith. Network dynamics and field evolution: The growth of interorganizational collaboration in the life sciences. *American J. of Sociology*, 110(4):1132–1205, 2005.

[27] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 105:1118–1123, 2008.

[28] S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[29] G. Simmel. *Conflict and the web of group affiliations*. Simon and Schuster, 1964.

[30] U. von Luxburg. A tutorial on spectral clustering. Technical Report 149, MPI for Biological Cybernetics, August 2006.

[31] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks *Nature*, 393:440–442, 1998.

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

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

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

[35] J. Yang and J. Leskovec. Structure and overlaps of communities in networks. In *SNAKDD '12*, 2012.

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