Finding Community Topics and Membership in Graphs

Matt Revelle, Carlotta Domeniconi, Mack Sweeney, and Aditya Johri

George Mason University,
Fairfax VA 22030, USA
{revelle,carlotta}@cs.gmu.edu,{msweene2,ajohri3}@gmu.edu

Abstract. Community detection in networks is a broad problem with many proposed solutions. Existing methods frequently make use of edge density and node attributes; however, the methods ultimately have different definitions of community and build strong assumptions about community features into their models. We propose a new method for community detection, which estimates both per-community feature distributions (topics) and per-node community membership. Communities are modeled as connected subgraphs with nodes sharing similar attributes. Nodes may join multiple communities and share common attributes with each. Communities have an associated probability distribution over attributes and node attributes are modeled as draws from a mixture distribution. We make two basic assumptions about community structure: communities are densely connected and have a small network diameter. These assumptions inform the estimation of community topics and membership assignments without being too prescriptive. We present competitive results against state-of-the-art methods for finding communities in networks constructed from NSF awards, the DBLP repository, and the Scratch online community.

1 Introduction

Given a graph of self-organizing objects, we wish to estimate the latent topics around which the objects organize and discover community membership. We hypothesize groups with high edge density in graphs are evidence of communities whose members have similar attributes within a subset of the feature dimensions.

In this paper we present Seeded Estimation of Network Communities (SENC). SENC is a probabilistic method which uses both node attributes and graph structure to simultaneously estimate community feature distributions and members. We assume a community may exist around seed groups in the network. Many community detection methods build strong assumptions regarding community features into their models, which limits generalizability. SENC provides a flexible means of accounting for a variety of community structures through the use of configurable lower and upper bounds on discovered communities. The seed groups define the lower bounds, and they may in turn be defined by network structure or node and edge attributes. In the experiments presented in

this paper, we consider every maximal k-clique in the network to be the core of a partially defined community. The upper bounds provide an intuitive way to incorporate knowledge about the degree of clustering in the network. Nodes may be members of multiple communities and communities may overlap.

Communities are defined by the associated distribution (topic) and a set of member nodes. Every seed group corresponds to a community, and the initial feature distributions are a weighted average of the seed members attributes. We use the features of nodes in each group to compute initial estimates for the community feature distributions (topics). We then find initial estimates of the membership weights given these estimated per-community topics. After this initialization, membership weights and community feature distributions are iteratively updated. The feature distributions are updated by aggregating attributes of community members and finding the maximum likelihood of a mixture distribution where the parameters for all other communities are fixed.

The contributions of this paper are:

- A scalable probabilistic method for simultaneously finding highly interpretable community topics and node memberships (SENC).
- A flexible and intuitive method of influencing community estimation through the use of bounded seed groups.
- The introduction of several datasets with ground-truth communities used for comparative experiments with top-performing methods.

2 Related Work

There are many approaches to community detection and the state-of-the-art methods which use both network structure and node features are based on linking models [21, 13], heuristic clustering [10, 11, 16], or topic models [15, 14]. Previous work [17] has also considered initialization with candidate communities.

Linking models estimate the probability of links and node attributes. They are similar to block models [2, 3] with link probabilities dependent on node attributes and community membership. Recent implementations are efficient and competitively find communities, but treat node and community features as binary values [21]. This results in a poor representation of the community's shared interest or topic.

There have been attempts at extending clustering methods to support network data, such as subspace clustering [10, 11, 16]. In contrast to linking models, these methods do not model edge probability and instead use observed edges and node attributes to identify dense, connected subgraphs with similar node attributes over a subset of the feature space. These methods are not probabilistic and rely on heuristics for detecting nodes with similar attributes. Further, they find many duplicates of a single detected community and require a distinct post-processing step to identify the optimal detected communities.

Topic model approaches extend basic models such as LDA [5] to estimate latent factors and introduce a dependence of edges on the latent factors. These models are generative and require a task-specific probabilistic graphical model.

In the past they have been difficult to scale up for larger datasets due to the sampling methods on which they rely [9].

3 Background

A substantial proportion of community detection techniques do not use node attributes to detect communities or provide per-community feature distributions as output. Many solely rely on graph structure [8] or independently group objects by topics and structure [23]. The state-of-the-art methods for community detection have introduced linking models, subspace clustering, topic models, and heterogeneous networks to improve performance and simultaneously estimate topics and membership.

The intuition of our model is most similar to subspace clustering and topic models and both are further discussed. We assume community members are similar across a subset of the feature space and we consider node feature values to be drawn from per-community feature distributions.

The recent literature on linking models which incorporate node attributes [21, 13] shows promising results. We aim to perform competitively with those methods by taking a different approach which is probabilistic but allows the use of heuristics to select seed groups.

Other literature [6] has focused on topic models for heterogeneous information networks. While our model is more general and does not require customization to support multiple types of nodes, we are still able to take advantage of the extra information provided by those networks by adding new features or edges.

3.1 Subspace Clustering

Subspace clustering is used to find clusters of objects that occur when the objects are embedded in a subset of the feature space dimensions. A survey of subspace clustering methods is provided in [12] which categorizes various approaches. Subspace clustering is frequently used on high-dimensional datasets and can be viewed as online feature selection for clustering [7].

A major challenge of subspace clustering is finding the optimal subspace clusters. A naive approach would exhaustively try every combination of features, but this is computationally infeasible for all but the smallest datasets. Our method is able to determine which features are relevant to each community by finding the maximal likelihood for the target community's feature distribution in the context of the mixture distribution which describes the node.

Our work extends research on subspace clustering in networks by introducing the use of probability distributions to describe the observed features and to estimate community topics and memberships. We view communities as having feature distributions which represent a common interest of all members.

3.2 Topic Models

Topic models are probabilistic models used to find the semantic structure of documents [4]. They are frequently generative and make assumptions about the relationships between topics, objects, and words. Some models support multiple topics per object or topic hierarchies, but the model is built with those assumptions. Topic models have been designed for networks which group related objects dependent on network structure [15].

The methods combining topic models with graph clustering tasks such as community detection are limiting. They either involve complex models which are only applicable to specific datasets or they independently find topics by treating vertices as documents and then attempt to fit the topics onto the graph to find clusters [23].

We represent node attributes as term-weight vectors and associate a topic with each community. Every cluster we find is a community, and each community has a single feature distribution or topic. We can then estimate a node's membership to a community by finding mixture weights which best explain the node's feature values through community topics.

4 Seeded Estimation of Network Communities

Network communities indicate interaction and attraction among members which is not shared by non-members. The nature of the interaction may be reflected in node attributes and we would expect for member nodes to be similar to one another. However, nodes may participate in multiple communities and the members of each community may be similar to each other in different ways.

To provide motivation for our method, let us discuss an example using an unspecified online social network. This social network allows users to join discussion areas for topics such as "computer science" or "coffee." Suppose a user is interested in both CS and coffee and participates in both communities. We expect the user's posts to the CS community will be different from her posts to the coffee community. We also expect the user's post in the CS community will be more similar in content to other posts in the CS community than to most posts in the coffee community.

Now assume we do not have access to individual user posts. Instead we have aggregated word counts for each user and we do not know which post contained which words. We can model a user's word frequencies as a random variable drawn from a multinomial distribution. Since each user may belong to different communities or have different levels of involvement then it's necessary to use a different multinomial distribution for each user. As previously hinted, we expect posts within a single community to have similar word frequencies. If we knew those per-community word distributions we could then represent each user's word distribution as a mixture distribution. This is akin to standard topic models such as LDA [4].

The Seeded Estimation of Network Communities (SENC) method described here has an advantage over state-of-the-art community detection methods in

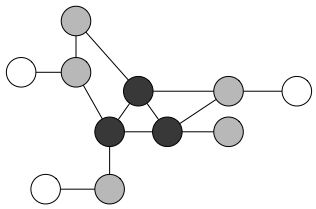

Fig. 1: Lower and upper bounds for a seed community. The lower-bound nodes are black, upper-bound nodes are grey, and excluded nodes are white.

its exploitation of network structure to regularize and guide estimation. This is possible through the use of *seed groups*. A seed group is a subgraph with properties which indicate the nodes are a subset of a community.

Each seed group is considered to be a lower bound of a community and its members are representative of this corresponding community. The lower-bound members, or *seed members*, influence estimation; the members' attributes are used as the initial estimate for the corresponding community's word distribution. The community topic is updated as additional member nodes beyond the lower bound are found.

Along with a lower bound, each seed community has a corresponding upper bound. The definition of this upper bound can be dependent on the network and its selection guided by simple network statistics such as the clustering coefficient. Figure 1 depicts an example of the bound sets for a seed community where the distance of a node from lower-bound members is used to define the upper-bound set. The bounds serve as a gentle bias to flexibly model assumptions regarding the shape of communities in a network.

Table 1: Definition of notation.

N	number of nodes
C	number of communities
D	number of feature dimensions
$\boldsymbol{\Phi}$	community topics, $C \times D$ matrix
$\boldsymbol{\Theta}$	community memberships, $N \times C$ matrix
$G(V, E)$	graph defined by vertices and edges
$S_{c=1\ldots C} \subseteq V$	members of community c
$\boldsymbol{x}$	attributes of a node

4.1 Notation

Before continuing it is useful to introduce notation and additional terms for describing the proposed method. We use *topic* to refer to the characteristic features of a community as well as the associated probability distribution parameters for all C communities, $\boldsymbol{\Phi}$, where each row $\boldsymbol{\Phi}_{c,*}$ is a parameter for the categorical

distribution associated with community $c = 1, \ldots, C$ with length D, the number of feature dimensions.

The node attribute vector $\boldsymbol{x}$ is a D-length vector of node feature values. A *membership weight vector* or *membership vector* is denoted as $\Theta_{n,*}$ and refers to the probability weight vector associated with node n over all C communities. The individual membership vectors make up the N rows of the membership matrix Θ. The membership weights indicate the proportion of node features which are attributed to each community. For quick reference, basic notation used in the equations is available in Table 1.

4.2 Model

SENC uses an EM algorithm to find the maximum-likelihood estimates for community topics and node memberships. Per-node community memberships are estimated as weighted counts of observed feature values given the community topics in the E-step and per-community topics are maximized in the M-step.

Node memberships for each node n participating in a seed group, $n \in \bigcup_{c=1\ldots C} S_c$, are estimated using the community topics. We represent the feature values of a node $\boldsymbol{x}$ as being drawn from a mixture distribution with per-node mixture weights $\Theta_{n,*}$ over all community distributions Φ using per-community topic distributions $\Phi_{c,*}$. A single term for a node n is drawn by first selecting a community c with probabilities $\Theta_{n,*}$ and then choosing a specific term with probabilities $\Phi_{c,*}$. For the data discussed in this paper, the community feature distributions are categorical distributions and node features $\boldsymbol{x}$ are generated by multiple trials of a mixture categorical distribution with proportions $\Theta_{n,*}\Phi$. A multinomial distribution is a categorical distribution with multiple, independent trials. We refer to the per-node feature distributions as multinomial distributions.

Nodes may be members of multiple communities and node features will then be characteristic of multiple community topics. In order to untangle the features characteristic of a community from those belonging to adjacent communities we define a mixture categorical likelihood function. This is the standard likelihood function but with the event probability vector $\boldsymbol{p}$ parameter computed as the matrix product of some $1 \times C$ mixture vector and $C \times D$ per-community topics matrix: $\Theta_{n,*}\Phi$.

We introduce $\boldsymbol{\gamma}$ as the sum of feature values from community members S_c to improve readability:

$$\boldsymbol{\gamma} = \sum_{n \in S_c} \boldsymbol{x} \tag{1}$$

When estimating $\Phi_{c,*}$ using community members S_c, the mixture vector $\boldsymbol{\theta}$ is a weighted average of membership vectors $\{\Theta_{n,*} : n \in S_c\}$ weighted by the proportional number of observations contributed by each node $n \in S_c$:

$$\boldsymbol{\theta} = \sum_{n \in S_c} \frac{(\sum_d x_d)\Theta_{n,*}}{\sum_d \gamma_d} \tag{2}$$

Using $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ we can now show how $\Phi_{c,*}$ may be updated. The event probability vector $\boldsymbol{p}$ is the parameter for a categorical distribution:

$$\boldsymbol{p} = \boldsymbol{\theta}\Phi \tag{3}$$

$$= \sum_{i=1}^{C} \theta_i \Phi_{i,*} \tag{4}$$

$$= \theta_c \Phi_{c,*} + \sum_{i=1, i \neq c}^{C} \theta_i \Phi_{i,*} \tag{5}$$

We can use the factoring of $\boldsymbol{p}$ in Equation (5) with the multinomial expected value to find the maximum-likelihood value of $\Phi_{c,*}$ given the community member observations $\boldsymbol{\gamma}$ from Equation (1).

The expected value for a single feature value i in random variable X drawn from $\text{Mult}(\boldsymbol{p}, n)$ is $E\{X_i\} = np_i$, where n is the number of trials and $\boldsymbol{p}$ is the event probability vector. If we replace the expected value of each feature dimension with the summation of community members' S_c attributes $\boldsymbol{\gamma}$ then we can substitute the expected value with the observed value γ_i for feature i and define:

$$\gamma_i = (\sum_{d=1}^{D} \gamma_d)\boldsymbol{\theta}\Phi_{*,i} \tag{6}$$

If we replace the expected value of each feature dimension with the summation of community members' S_c attributes $\boldsymbol{\gamma}$ then we can define the maximum likelihood of $\Phi_{c,*}$ as:

$$\boldsymbol{\gamma} = (\sum_d \gamma_d)\boldsymbol{\theta}\Phi \tag{7}$$

$$= (\sum_d \gamma_d)(\theta_c \Phi_{c,*} + \sum_{i=1, i \neq c}^{C} \theta_i \Phi_{i,*}) \tag{8}$$

$$\frac{\boldsymbol{\gamma}}{\sum_d \gamma_d} = \theta_c \Phi_{c,*} + \sum_{i=1, i \neq c}^{C} \theta_i \Phi_{i,*} \tag{9}$$

$$\theta_c \Phi_{c,*} = \frac{\boldsymbol{\gamma}}{\sum_d \gamma_d} - (\sum_{i=1, i \neq c}^{C} \theta_i \Phi_{i,*}) \tag{10}$$

$$\Phi_{c,*} = \frac{\frac{\boldsymbol{\gamma}}{\sum_d \gamma_d} - (\sum_{i=1, i \neq c}^{C} \theta_i \Phi_{i,*})}{\theta_c} \tag{11}$$

Using Equation (11) we can easily estimate community topics using node attributes, per-node community membership weights, and the latest topic estimates for other communities.

We use Φ' to reference a modified version of Φ with normalized columns, each summing to 1. The per-node community memberships are found by performing a weighted count of node attributes over the communities, where α denotes a normalization scalar:

$$\Theta_{n,*} = \alpha \Phi' \boldsymbol{x}^T \tag{12}$$

For each observed term, we assign a proportion of the count to each community according to the relative probability of that term occurring in each community. A community with a higher relative probability of a given term occurring will receive a larger proportion of the count than the others.

4.3 Algorithm

The SENC algorithm constructs per-community lower- and upper-bound matrices, initializes per-community topics Φ and per-node community memberships Θ, and then performs expectation-maximization iterations until estimates stop improving or the maximum number of iterations is reached. The algorithm requires the $N \times N$ graph adjacency matrix and the $N \times D$ node attribute matrix as input. The lower-bound matrix is a $C \times N$ binary matrix of the seed members where 1-values indicate node n belongs to community c. The upper-bound matrix is a binary $N \times C$ matrix where 1-values indicate node n may belong to community c. This prevents nodes from distant communities being assigned to communities with a similar topic. The construction of lower- and upper-bound sets for each community is dependent on the network being processed. Two matrices are produced as output: a $C \times D$ community topic matrix and an $N \times C$ community membership matrix. A goal of our method is to remove features representative of overlapping communities over EM iterations. The node membership vectors are estimated using the community topics to perform a weighted count over node attributes. These weighted counts are normalized to sum to one and used as membership weights.

Algorithm 1 Main Program: initialization, EM, termination.

Input: The *graph* and *node attributes*.
Output: The *community topics* and *membership*.

```
Construct lower-bound and upper-bound matrices;
Initialize community topics Φ and memberships Θ;
while Not convergent or max iteration do
    Call E-step to update membership Θ;
    Call M-step to update community topics Φ;
    Check for convergence;
end while
return Community topics Φ and membership Θ;
```

Algorithm 2 E-step: update per-node community memberships.

Input: The *community topics*, *upper-bound matrix*, and *node attributes*.
Output: The updated *membership*.

for Each node n **do**
    Identify which communities influence node n;
    Select topics of influential communities;
    Compute weighted counts from selected topics with Equation (12);
    Assign normalized counts to membership vector $\Theta_{n,*}$;
end for
return Updated membership Θ;

Algorithm 3 M-step: update per-community topics.

Input: The *node attributes*, *membership*, *influence*, and *community topics* from the previous iteration.
Output: The updated *community topics*.

for Each seeded community c **do**
    Select all nodes with membership in c;
    Compute weighted average of selected nodes' membership by Equation (2);
    Estimate topic with Equation (11);
    Assign updated topic to parameter vector $\Phi_{c,*}$, if likelihood improves;
end for
return Updated topics Φ;

After initial estimates are calculated, the algorithm alternatively updates the node memberships and community topics. The per-node and per-community iterations within the E- and M-step are independent and computation may be distributed across multiple threads. The E-step in Algorithm 2 updates the per-node community memberships for all nodes given the community topics, influence matrix, and node attributes. This is done by computing the weighted counts of node attributes using the probability of each attribute for each community, as shown in Equation (12). The upper-bound complexity of the E-step is $O(N\mathcal{C}\mathcal{D})$, where $\mathcal{C}$ and $\mathcal{D}$ are the number of communities to which a node may belong and the number of dimensions relevant to those communities. In practice, $\mathcal{C}$ and $\mathcal{D}$ will be much smaller than C and D.

The M-step, shown in Algorithm 3, updates the per-community feature distributions. We find a new estimate for $\Phi_{c,*}$ using Equation (11) and compare its log-likelihood to the previous iteration's estimate. The new estimate is used if it better explains the feature values of the member nodes. The M-step has computational complexity of $O(C(\mathcal{N}\mathcal{C}+\mathcal{N}\mathcal{D}+\mathcal{C}\mathcal{D}))$, where $\mathcal{N}$ is the number of nodes in the upper-bound set of a community, $\mathcal{C}$ is the number of communities associated with the $\mathcal{N}$ nodes, and $\mathcal{D}$ is the number of feature dimensions relevant to all $\mathcal{C}$ communities and $\mathcal{N}$ nodes. Again, $\mathcal{C}$, $\mathcal{N}$, and $\mathcal{D}$ are usually much smaller than C, N, and D.

5 Experiments

We evaluate our proposed method on networks with varying structure to determine whether SENC's results are consistently competitive with state-of-the-art methods. The networks considered are: an NSF research collaboration network, several DBLP citation networks [19], and a Scratch project collaboration network. For comparison, we evaluate the performance of four state-of-the-art community detection methods: CESNA [21], CoDA [22], EDCAR [10], and Link Clustering [1]. CESNA and EDCAR use network structure and node attributes to detect communities; however, the current implementations struggled to process networks with a large number of features. In order to evaluate more methods we elected to use smaller datasets. CoDA and Link Clustering only use network structure.

An implementation of SENC and datasets used in experiments will be made available at the GMU DMML website[1].

5.1 Dataset Descriptions

We construct a research collaboration network from NSF awards granted by the Directorate for Computer and Information Science and Engineering (CISE) between January 1995 and August 2014. This is accomplished by forming undirected edges between the PI and co-PIs who received funding from the same award. The awards are associated with programs and we use the programs with at least three associated researchers as ground truth. We find 90% of researchers received funding from six or fewer programs; this suggests programs function well as ground-truth communities. There are a total of 768 programs in the CISE Directorate. NSF awards data is publicly available from the NSF website[2].

An online computer science bibliography, DBLP, contains entries for published papers with information about the authors, citations, and publication venues. The per-year DBLP citation networks were constructed from an existing citation dataset [19] by forming edges between authors who cited each other within that year. Papers are linked to a publication venue and these venues were used to define ground truth. Venues referenced only once were removed from our dataset. Venues with three or more associated authors were used as ground truth.

Scratch [18] is an online community where users may write and share projects (programs) with other users. One way in which Scratch users may interact is by remixing projects. Remixing allows a user to create a copy of any existing project which they may then modify. We created a co-remix affiliation network from the MIT Scratch Team's dataset containing users, projects, and remixes. An edge is formed when two users remix the same project. To reduce the total number of edges we used co-remix edges where users had three or more projects in common. Users may create project galleries which are curated collections of

[1] http://cs.gmu.edu/∼dmml
[2] http://www.nsf.gov/awardsearch/download.jsp

projects. Galleries corresponding to three or more users were used as ground truth. The Scratch dataset used to construct the network may obtained from the MIT Media Lab website[3].

Table 2: Network statistics. N: number of nodes, E: number of edges, D: number of node attributes, MC: number of maximal cliques with 3+ members, GCC: global clustering coefficient, LCC: average local clustering coefficient, G: number of ground-truth communities.

Dataset	N	E	D	MC	GCC	LCC	G
NSF	8,168	38,212	43,445	3,331	0.590	0.683	429
DBLP 2010	32,961	130,420	58,007	37,120	0.422	0.440	2,288
DBLP 2011	32,614	131,921	56,166	39,955	0.421	0.438	2,215
DBLP 2012	33,576	135,883	54,269	42,443	0.381	0.397	1,861
Scratch	1,714	17,824	36,494	7,705	0.584	0.704	718

Several of the methods make use of node attributes and these were provided as tf-idf weighted values for EDCAR and SENC and binary values for CESNA. For the NSF CISE network, terms associated with each researcher were taken from NSF award titles and abstracts. The DBLP author terms were taken from titles and abstracts of papers they wrote. Scratch user terms were extracted from titles, descriptions, and tags of their projects. The term features in all networks had stop words removed and terms stemmed.

Multiple connected components were found in all networks and the smaller components were removed as they may be trivially considered communities. Table 2 lists the network statistics for the largest component of each network used for experiments and analysis. All the networks used for experiments are undirected, but they vary in structure.

As shown in Table 2, the NSF and Scratch networks have higher clustering coefficients than the DBLP networks. This is unsurprising as the NSF and Scratch networks are affiliation networks (co-award and co-remix). Our experiments show that while SENC is able to perform competitively across all the networks other methods tend to either perform better on networks with higher or lower clustering coefficients.

5.2 Methods and Evaluation

The public implementations of CESNA, CoDA, EDCAR, and Link Clustering were used. CESNA and CoDA rely on an estimate of the number of communities. We provided the number of NSF programs, DBLP publication venues, and Scratch galleries as estimates. CoDA is designed for directed networks but can be used to find communities in undirected networks. It does this by processing the network twice, switching the direction of edges between runs. As a result,

[3] https://llk.media.mit.edu/scratch-data

two sets of detected communities are generated. We combined both sets when evaluating the performance of CoDA. EDCAR requires 10 parameters and the suggested values from the implementation documentation were used. Link Clustering is parameter-less and only requires the edge list as input. Maximal cliques of size three and above were used as the lower-bound groups for SENC and the upper-bound groups were selected based on the clustering coefficient. The high clustering coefficients of the NSF and Scratch networks indicate tighter upper bounds should be used than with the DBLP networks. For the DBLP networks we extend the lower bounds by including all nodes adjacent to any lower-bound member. The upper bounds for the NSF and Scratch networks are simply the same maximal cliques.

Link Clustering and SENC require a post-processing step to define exact communities. The Link Clustering implementation includes a script to calculate the optimal dendrogram cut threshold and we use this to determine the communities for evaluation. SENC defines community membership with probabilities and does not perform a hard assignment of nodes to communities like the other evaluated methods. We account for this in our evaluation by filtering weaker memberships. For all nodes, we sort their memberships in descending order by weight and take all the assignments until the sum of weights reaches a minimum threshold value. An optimal threshold is used for each dataset.

We use the evaluation function described in [21, 20] and recited in Equation (13) to compute the $F1$ score and Jaccard similarity of detected communities against ground-truth communities. This function is especially useful when the numbers of detected communities and ground-truth communities differ as occurs with several of the methods in our experiments. In Equation (13), C^* denotes a set of ground-truth communities, C a set of detected communities, and $\delta(\cdot)$ is a similarity metric.

$$\frac{1}{2|C^*|}\sum_{C_i^* \in C^*}\max_{C_j \in C}\delta(C_i^*, C_j) + \frac{1}{2|C|}\sum_{C_j \in C}\max_{C_i^* \in C^*}\delta(C_i^*, C_j) \tag{13}$$

5.3 Results

Using the evaluation function defined in Equation (13) we find the $F1$ score and Jaccard similarity between the detected communities from all methods and the ground-truth communities.

Table 3: $F1$ scores for all methods and datasets.

Method	Attr.	NSF	DBLP10	DBLP11	DBLP12	Scratch	Avg.
CoDA	No	0.216	0.278	0.273	0.263	0.283	0.263
Link Clust.	No	0.303	0.266	0.265	0.258	**0.399**	0.298
CESNA	Yes	0.228	0.272	0.263	0.255	0.356	0.275
EDCAR	Yes	0.164	N/A	N/A	N/A	N/A	N/A
SENC	Yes	**0.346**	**0.301**	**0.297**	**0.298**	0.365	**0.321**

Table 4: Jaccard index for all methods and datasets.

Method	Attr.	NSF	DBLP10	DBLP11	DBLP12	Scratch	Avg.
CoDA	No	0.132	0.172	0.168	0.162	0.174	0.162
Link Clust.	No	0.233	0.166	0.166	0.161	**0.265**	0.198
CESNA	Yes	0.139	0.167	0.161	0.156	0.228	0.170
EDCAR	Yes	0.112	N/A	N/A	N/A	N/A	N/A
SENC	Yes	**0.269**	**0.190**	**0.187**	**0.190**	0.235	**0.214**

Our results are provided in Tables 3 and 4 and show SENC outperforms most other methods over all datasets and achieves the highest average performance. Unfortunately, the current implementation of EDCAR was unable to process most of the networks. We believe this is partly due to the large number of features.

We note the relative difference in performance of CoDA and CESNA to Link Clustering flips between the networks with higher and lower clustering coefficients. In the NSF and Scratch networks, Link Clustering outperforms CoDA and CESNA but performs worse than CESNA on the DBLP10 network and worse than CoDA on every DBLP network. This may indicate these other methods include a biased definition of communities which is not found in all social networks. SENC performs well across all the networks and avoids this problem through the use of its configurable bounds chosen based on network statistics such as clustering coefficients.

5.4 Interpretation of Detected Communities

We also perform a qualitative analysis on communities discovered by SENC to illustrate the interpretability of its results. Several communities relating to data mining and machine learning were found in the NSF CISE network.

Table 5: Top-5 researchers of the AMPLab and Computational Learning communities with corresponding membership weights.

AMPLab		Comp. Learning	
Peter Bartlett	0.5084	Laurent El Ghaoui	0.5884
Laurent El Ghaoui	0.4116	Peter Bartlett	0.4916
Michael Franklin	0.1346	Jesse Snedeker	0.4647
Michael Jordan	0.1049	Federico Girosi	0.4134
Alexandre Bayen	0.0996	Robert Berwick	0.2830

We present the top-5 researchers and top-40 terms of two such groups in Table 5 and Figure 2. The first community is associated with Berkeley's AMPLab[4], which works on problems involving machine learning, cloud comput-

[4] https://amplab.cs.berkeley.edu

Fig. 2: Word clouds of the top-40 terms from the AMPLab community (left) and computational learning community (right).

ing, and crowdsourcing. The top-5 researchers are all EECS faculty at Berkeley and Michael Franklin and Michael Jordan are both directors of AMPLab. Recall membership weights are normalized per-researcher and a lower membership weight indicates the researcher's work is also captured by other community topics. Most of the terms are self-explanatory, but the term *Alon* refers to Alon Halevy of University of Washington whose name appears in several award abstracts and has collaborated with Michael Franklin.

We find another community with 12 members in common with the AMPLab community. Its topic may be described as computational learning and its applications to computer vision and natural-language processing. The AMPLab and computational learning communities have 41 and 34 members respectively, with roughly about one-third being shared. These common members include: Michael Jordan, Michael Franklin, Peter Bartlett, and Tomaso Poggio.

Although both communities are generally concerned with human-centric applications of machine learning, the AMPLab community is focused on computing architecture to solve such problems, while the computational learning community is focused on understanding human vision and motor control. This discovery of overlapping communities with shared general interests but distinct features exemplifies an advantage of SENC's initialization by seed groups.

6 Conclusion

We have introduced SENC — a probabilistic approach to community detection that outputs node memberships and community topics. Simple network statistics, such as the clustering coefficient, can be used to guide configuration of flexible bounds on seed groups. The bounded seed groups enable SENC to account for differences in underlying community structure across many networks. This contrasts with existing methods which build strong assumptions into their models. As a result, SENC is able to consistently outperform state-of-the-art community detection methods on a variety of networks. No other method performed consistently across all the networks used in our experiments. This indicates SENC generalizes better than current state-of-the-art methods.

The output produced by SENC is highly interpretable. We can understand the nature of a discovered community by examining its topic distribution. We can also review a node's relative community involvement through its membership weights. The combination of SENC's flexible model and interpretable results make it an excellent choice for both exploratory analysis of networks and community detection tasks.

Our experiments have raised several interesting questions for future work. We are interested in discovering how network characteristics affect assumptions made in community detection methods and how other approaches for defining bounded seed groups may further improve SENC's performance.

7 Acknowledgment

We appreciate the Lifelong Kindergarten group at MIT for publicly sharing the Scratch datasets. This work is partly based upon research supported by U.S. National Science Foundation (NSF) Awards DUE-1444277 and EEC-1408674. Any opinions, recommendations, findings, or conclusions expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

References

1. Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.
2. E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.
3. R. Balasubramanyan and W. W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *Proceedings of the SIAM International Conference on Data Mining*, volume 11, pages 450–461. SIAM, 2011.
4. D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
6. H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1271–1279. ACM, 2011.
7. C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *Proceedings of the SIAM International Conference on Data Mining*, pages 517–521. SIAM, 2004.
8. S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
9. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, 1984.
10. S. Günnemann, B. Boden, I. Färber, and T. Seidl. Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors. In *Advances in Knowledge Discovery and Data Mining*, pages 261–275. Springer, 2013.

11. S. Günnemann, I. Färber, B. Boden, and T. Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *Proceedings of the IEEE International Conference on Data Mining*, pages 845–850. IEEE Computer Society, 2010.
12. H.-P. Kriegel, P. Kröger, and A. Zimek. Subspace clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):351–364, 2012.
13. J. Leskovec and J. McAuley. Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems*, pages 539–547, 2012.
14. Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link LDA: Joint models of topic and author community. In *Proceedings of the International Conference on Machine Learning*, pages 665–672. ACM, 2009.
15. A. McCallum, X. Wang, and N. Mohanty. Joint group and topic discovery from relations and text. In E. Airoldi, D. Blei, S. Fienberg, A. Goldenberg, E. Xing, and A. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, volume 4503 of *Lecture Notes in Computer Science*, pages 28–44. Springer Berlin Heidelberg, 2007.
16. F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *Proceedings of the SIAM International Conference on Data Mining*, volume 9, pages 593–604. SIAM, 2009.
17. S. Pool, F. Bonchi, and M. v. Leeuwen. Description-driven community detection. *ACM Transactions on Intelligent Systems and Technology*, 5(2):28:1–28:28, 2014.
18. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al. Scratch: Programming for all. *Communications of the ACM*, 52(11):60–67, 2009.
19. J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 990–998, 2008.
20. J. Yang and J. Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 587–596, New York, New York, USA, 2013. ACM.
21. J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *IEEE 13th International Conference on Data Mining*, pages 1151–1156. IEEE, 2013.
22. J. Yang, J. McAuley, and J. Leskovec. Detecting cohesive and 2-mode communities in directed and undirected networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 323–332. ACM, 2014.
23. Z. Zhao, S. Feng, Q. Wang, J. Z. Huang, G. J. Williams, and J. Fan. Topic oriented community detection through social objects and link analysis in social networks. *Knowledge-Based Systems*, 26:164–173, 2012.