On the Separability of Structural Classes of Communities

Bruno Abrahao Sucheta Soundarajan John Hopcroft Robert Kleinberg

Department of Computer Science
Cornell University
Ithaca, NY, 14850, USA
{abrahao, sucheta, jeh, rdk}@cs.cornell.edu

ABSTRACT

Three major factors govern the intricacies of community extraction in networks: (1) the application domain includes a wide variety of networks of fundamentally different natures, (2) the literature offers a multitude of disparate community detection algorithms, and (3) there is no consensus characterizing how to discriminate communities from non-communities. In this paper, we present a comprehensive analysis of community properties through a class separability framework. Our approach enables the assessment of the structural dissimilarity among the output of multiple community detection algorithms and between the output of algorithms and communities that arise in practice. To demonstrate this concept, we furnish our method with a large set of structural properties and multiple community detection algorithms. Applied to a diverse collection of large scale network datasets, the analysis reveals that (1) the different detection algorithms extract fundamentally different structures; (2) the structure of communities that arise in practice is closest to that of communities that random-walk-based algorithms extract, although still significantly different from that of the output of all the algorithms; and (3) a small subset of the properties are nearly as discriminative as the full set, while making explicit the ways in which the algorithms produce biases. Our framework enables an informed choice of the most suitable community detection method for a given purpose and network and allows for a comparison of existing community detection algorithms while guiding the design of new ones.

Categories and Subject Descriptors

I.5.1 [**Computing Methodology**]: Pattern Recognition — *Design Methodology*

Keywords

Networks, Community Structure, Detection Algorithms, Class Separability

KDD'12, August 12–16, 2012, Beijing, China.

1. INTRODUCTION

Community structure captures the tendency of entities in a network to group together in meaningful subsets whose members have a distinctive relationship to one another. The identification of these subsets allows for the analysis of networks at different levels of detail, which is instrumental in illuminating the structure underlying large-scale systems [5, 9, 10, 22, 23].

Despite playing a fundamental role in the structure and function of networks, community structure has proved to be frustratingly difficult to define, quantify, and extract. In addition to challenges related to computational tractability, three major factors account for the intricacies of community extraction. First, the application domain includes a wide variety of networks of fundamentally different natures. Each of these networks possesses meaningful communities that may possess their own distinctive structural profiles. Second, the literature offers a multitude of disparate community detection algorithms. Due to differences in concept and design, the output of these procedures exhibits high structural variability across the collection. Last, there is no established consensus on the question of what properties distinguish subgraphs that are communities from those that are not communities.

In this paper, we tackle these challenges through a comprehensive analysis of community properties. We present a framework that enables researchers and practitioners to assess the structural dissimilarity among the output of multiple community detection algorithms and between the output of algorithms and communities that arise in practice. Our approach analyzes communities by taking account of a broad spectrum of structural properties. The analysis reveals nuances of the structure of real and extracted communities.

We frame our approach as a class separability problem, which simultaneously handles many classes of communities and a diverse set of structural properties. To this end, we specify a learning problem in which we map the distinct communities into a feature space, where the dimensions represent measures that characterize a community's link structure. The separability of classes provides information on the extent to which different communities come from the same (or fundamentally different) distributions of feature values. We extract different classes of communities that can be grouped into two categories: intrinsically-defined and extrinsically-defined communities.

We define the first set of communities by properties intrinsic to their link structure. For our purposes, these are the sets that community detection algorithms may output.

Each class of intrinsically defined communities comprises a set of examples that a specific algorithm extracts. The separability of these classes demonstrates the extent to which different algorithms output structurally distinguishable subgraphs. A feature selection analysis can then be employed to highlight the properties that exhibit the highest degree of inter-class variability, thereby making explicit the structural bias produced by different algorithms.

We also define communities by the context, the dynamics, or the function associated with the networks, but extrinsic to the link structure. We identify these communities through meaningful annotations provided with the datasets, such as explicit declaration of membership, product categories, grouping by protein function, and so on. In this fashion, for each network, we form a class of extrinsically-defined communities, henceforth called *annotated communities*. These communities enable a large-scale rigorous analysis of community detection methods. The separability of the class comprising annotated communities from the classes of intrinsically-defined communities determines the extent to which community detection algorithms succeed in extracting subgraphs that are structurally comparable to the communities formed by nodes sharing extrinsic properties in common.

To demonstrate our approach, we furnish our framework with a large set of structural properties and ten different community detection procedures to produce examples of different structural classes. Our selection is representative of categories of popular algorithms available in the literature. We consider a diverse collection of large scale real networks whose domains span biology, on-line shopping, and social systems. Assessing separability using supervised classifiers both parametric, namely Support Vector Machines [30], and nonparametric, namely k-Nearest Neighbors [1], together with a feature selection analysis using correlation-based methods [11], we reach the following conclusions about the communities in question. First, for all networks, the strong cross validation performance indicates that the different community detection algorithms produce fundamentally different structures that are separable on the feature space defined. Second, we observe that in nearly all cases, the annotated communities are structurally distinguishable from the output of all community detection algorithms. Nevertheless, the communities bearing the closest structural resemblance to annotated communities are those that random-walk-based algorithms extract. Surprisingly, in spite of the diversity of the domains from which our networks are drawn, this observation applies to all of the networks, except to two of them for which we have a small population size. Finally, a small subset of the features is consistently observed as the most discriminative. This observation allows for a dimensionality reduction by a factor as large as 4, preserving an equivalent 10-fold cross validation performance. The most discriminative features identify the ways in which the different algorithms produce biases. As illustrated by our experiments, by producing artificial or real examples of communities that possess the structure we wish to find, we can use our framework to enable an informed choice of the most suitable community detection method for a given network. In addition, it allows for a comparison of existing community detection algorithms and may guide the design of new ones.

This paper is organized as follows. Section 2 discusses background information and related work. Section 3 introduces the datasets we use, the algorithms we consider, and the measures we apply to construct the feature space. Section 4 describes the heart of our framework and presents an experimental analysis thereof. Next, this section closes with a feature selection analysis. Finally, Section 5 offers our concluding remarks.

2. BACKGROUND AND RELATED WORK

The work by Girvan and Newman [10] has recently sparked a wave of interest in the notion of *community structure* as a decomposition of a network that reflects meaningful properties of the underlying system [9]. Nevertheless, this area has its roots in the related problem of graph partitioning whose initial contributions date back to the 1970s [14].

Given the diverse nature of networks, the notion of meaningful communities is necessarily context dependent, involving interpretations and expectations of domain experts. Therefore, many attempts to define communities are grounded on the notion of mathematical optimization. Starting with an a priori expectation about what a community should look like, researchers specify an objective function for a search method, whose solution for a given instance provides the desired communities. This process has given rise to a a large collection of community detection algorithms, which aim at optimizing various objective functions. As mentioned in the previous section, the multitude of community structure definitions represents a source of high variability between the output of different community detection algorithms.

Among the objective functions introduced in previous work, the notion of *modularity* [10] has become an influential one. Modularity assigns high scores to communities whose internal edges outnumber the ones established in expectation by a random-network model that preserves the degree distribution of the original network. A related notion inspired by electrical networks is that of *conductance* [5]. The conductance of a set S with complement S^C is the ratio of the number of edges connecting nodes in S to nodes in S^C by the total number of edges incident to S or to S^C (whichever number is smaller). The common theme underlying the preceding notions is the search for node sets that are internally cohesive and yet sparsely connected to the rest of the network. Therefore, these measures tend to penalize sets having a large number of edges crossing the set relative to the count of internal edges.

Communities in general, however, display features that modularity and conductance may not capture, such as a preponderance of links to the outside over internal links and an arbitrary degree of overlap. This fact is substantiated by an investigation of real networks revealing that they do not split well into low-conductance communities [17] as most networks are expander-like [12]. These considerations lead to the development of alternative definitions, such as (α, β)-*community* [20], and algorithms, such as *Link Communities* [2] and *Clique Percolation* [24].

Communities in real networks often emerge as a result of multiple driving forces that make up the underlying complex system. Therefore, the attempt to capture community structure by maximizing a given objective function may represent an unrealistic expectation. As a consequence, communities identified by methods that reflect mathematical constructs may differ structurally from real communities that arise in practice. Despite the vast literature on community detection, the work by Ahn et al. [2], as well as ours, are among the few that attempt to analyze the structural

resemblance between communities extracted by algorithms and annotated communities, which represent examples of meaningful communities in various domains.

Even though network analysts expect the outputs of the different algorithms to display dissimilar structural profiles due their conceptual diversity, the structural variability does not hinge simply on the choice of optimization problem. In most cases of interest, the search for a collection of node sets that maximize a given objective function is computationally intractable [9]. Therefore, in an attempt to handle the massive scale of today's networks, popular methods of community detection rely on efficient heuristics. As a consequence, previous work have quantified a significant output variability among different approximation algorithms that aim at maximizing the exact same function [15, 18].

In the spirit of studying the structural variability exhibited by different algorithms, closest to ours is the work by Leskovec et al. [17], which discusses properties of communities produced by multiple algorithms that aim at maximizing conductance. They consider the values of a handful of features, e.g., set compactness and internal conductance, produced by different algorithms. In contrast, here we present the first study that is simultaneously comprehensive with respect to the diversity of structural properties, of domains, of algorithms, and of scale. We take account of a set of 36 features, measured from the output produced by 10 different community detection processes, which are representative of classes of available algorithms that aim at maximizing various different objective functions in the literature. We derive our results from a diverse collection of datasets from large-scale networks arising from multiple domains.

3. BUILDING STRUCTURAL CLASSES

Before describing our framework and delving into our analysis, in this section we present the datasets we use, as well as our methodology for building structural classes of communities from the network data. We also define the feature space for our learning problem.

3.1 Datasets

We analyze eight large scale datasets, namely DBLP, LiveJournal, two portions of the Facebook network (denoted by Facebook — Rice University Undergraduate and Graduate), Amazon, and three biological networks denoted by HS, SC, and Fly. The collection encompasses different forms of entities and relationships and originate from diverse domains.

The LiveJournal dataset consists of a snapshot of a large network of bloggers, previously explored in [3]. The snapshot includes 4,847,571 bloggers who explicitly declare their friendship links. Due to the massive size of this dataset, we consider two portions of it, which we obtain by starting at a random node and performing a breadth-first search from that node. The datasets, henceforth named LJ1 and LJ2, contain 500,000 nodes each. LJ1 and LJ2 contain 10,736,588 and 10,640,429 edges, respectively.

DBLP data is publicly collectible and our dataset consists of a snapshot taken in May 2009 of the on-line publications database site DBLP. The data include a collection of editions of publication venues (i.e., conferences and journals) in computer science. A pair of the 744,386 authors present in the dataset are linked if they have co-authored at least one paper in any of the venues.

Facebook — Rice University Undergraduate (Ugrad) and Graduate (Grad) are an anonymized portion of the Facebook network which include Rice University students, collected by crawling public friends lists on Facebook on May 17, 2008. They consist of two disjoint sets of 1220 undergraduate students and 503 graduate students, respectively. Mislove et al. [21] present a detailed description of these datasets.

The Amazon dataset [16] is a product co-purchasing network from the on-line retailer Amazon.com. Each node represents a book, and an edge exists between two nodes if one was frequently purchased with the other. The network contains 270,347 nodes and 741,142 edges. For each book, Amazon.com reports up to 5 other items that were frequently purchased with the book.

Biological networks HS, SC, and Fly describe protein-protein interactions for *H. Sapiens* (human), *S. Cerevisiae* (a type of yeast), and *Drosophila* (a fruit fly species) [25], respectively. In these networks, a node represents a protein, and two nodes are connected if scientific evidence of their interaction exists. HS contains 10,298 nodes and 54,655 edges, SC contains 5523 nodes and 82,656 edges, and Fly contains 15,326 nodes and 486,970 edges.

3.1.1 Annotated communities

The networks we analyze contain annotations reflecting examples of communities that arise in these domains[1]. Some of these sets are user-defined, i.e., users explicitly declare their participation in the community, while others reflect contextual information of the underlying process or organization, e.g., department, protein function, product category, etc.. Below we describe how we identify and clean the annotated communities for each dataset.

For the social networks, in LiveJournal, users explicitly declare their membership in zero or more communities created and administered by users. In DBLP, conferences where authors publish their research work reflect the community memberships. Finally, for Facebook — Rice University Undergraduate and Graduate, users who possess common academic attributes, such as department, major, or dormitory, form the communities. These attributes were obtained by matching Facebook names with student records from the university's directory [21].

For each item in Amazon.com, the on-line store provides several product categories, such as "Photo Essays" or "Landscape Architecture Textbooks". We identify a set of nodes possessing a common categorical label as a community.

For HS, SC, and Fly, a number of proteins (though not all) have annotations regarding one or more gene ontology IDs describing the known functions that the protein serves (e.g., metabolic regulation). We use these gene ontology values to identify the communities.

Since we define annotated communities extrinsically to the link structure, the sets formed by the preceding definitions may induce disconnected graphs. However, it is reasonable to limit the definition of community to include only connected subgraphs of the network. Therefore, to capture the community information implicit in the annotations, we consider each connected component of the graph induced by a node set possessing a common label as an annotated community by itself. Moreover, since we are interested in the

[1]These communities, however, may not represent an unbiased sample of communities in these networks as other communities that are not annotated might also exist.

structure of reasonably sized communities, we filtered out small communities with less than 10 members.

Overall, we identified 29,955 annotated communities for LJ1, 39,598 for LJ2, 10,595 for DBLP, 24 for RICE-grad, and 41 for RICE-ugrad, 9439 for Amazon, 64 for HS, 76 for SC, and 54 for Fly.

3.2 Structural classes and feature space

In this section we describe how to produce examples that constitute the structural classes and how to build the feature space for our learning framework. The process consists of two steps. First, we produce the examples by applying community detection algorithms, one for each class, to the network data. Second, we extract features by measuring a broad spectrum of properties of the subgraphs induced by communities. This latter step uses a set of examples consisting of the output produced in the previous step along with the set of annotated communities.

3.2.1 Producing the examples

To study classes of intrinsically defined communities, we selected a collection of 10 community detection procedures, which are representative of strategies employed by a broad range of algorithms in the literature. We applied these procedures to each of the nine network datasets to extract examples of subgraphs produced by these methods. We labeled examples with the identity of the community detection procedure that produced them. In total, for each network, we created 11 structural classes of communities: one class of extrinsically-defined communities, which comprises examples of annotated communities, and each of the other 10 classes corresponding to intrinsically-defined communities, which comprise examples extracted by each of the 10 community detection algorithms respectively. Below we briefly describe the community detection procedures we consider.

1. **Breadth First Search (BFS):** To establish a baseline, we use breadth first search to extract sets that serve as examples of random connected communities. To create one BFS community of size k, we begin with a randomly selected node and perform a breadth first search from that node until we visit k elements.

2. **Random Walk 0 (RW0):** The central idea in many community detection algorithms is that random walks tend to concentrate within a community [26, 31]. To create communities of size k, we begin with a random node and perform a uniform–random walk from that node until k different nodes are visited.

3. **Random Walk 0.15 (RW15):** This is similar to the preceding method with the twist that at each step we restart the walk from the starting node with 0.15 probability. RW15 concentrates the random walk distribution around a center, thereby forming more compact sets, whereas RW0 communities tend to spread out.

4. (α, β) **(AB)**: An (α, β)-community, for $\alpha < \beta$, requires every member of the community to be connected to at least β other members while nonmembers have at most α links to the community [20]. This definition allows for overlapping communities whose out-links may outnumber the in-links. To produce an AB community of size k, we produce a BFS community of size k and then apply a limited number of sequential node swaps that aim at making the set an approximate or exact (α, β)-community. In each step we remove the community node with the fewest member neighbors and add the fringe node with the most member neighbors.

5. **Link Communities (LC):** In contrast with the majority of the available algorithms, Link Communities [2] aims at addressing the overlapping and hierarchical nature of community structure by treating communities as groups of links rather than nodes. We extract examples of this structure by applying a standard implementation of this algorithm to our networks.

6. **Infomap (IM):** The Infomap algorithm [27] views the problem of finding communities as akin to the problem of a map-maker deciding on a level of granularity. The communities and the nodes therein have names. A random walk in the network is described by appending the community name followed by the name of nodes visited while in the community to a transcript. The goal is to find the community structure that minimizes the expected length of the description. Intuitively, such a structure would cause random walks to rarely escape communities.

7. **Louvain:** The Louvain method [4] is a popular method for greedy modularity optimization. The algorithm consists of iteratively aggregating nodes into communities whenever this move locally improves modularity. The process outputs communities when no further merge produces a significant gain in modularity.

8. **Newman-Clauset-Moore (Newman):** This method is another example of greedy modularity maximization [6]. Unlike the Louvain method, which considers merges that locally improve modularity, Newman-Clauset-Moore identifies a hierarchical community structure from which communities are extracted by cutting the dendrogram that reflects the hierarchy at the level that maximizes a global value of modularity.

9. **Markov Clustering Algorithm (MCL):** MCL [7] is a random-walk-based method. It consists of two alternating steps. It begins with the random-walk matrix of a graph (the normalized adjacency matrix). The first step, namely "expansion", squares this matrix; this corresponds to computing the flow between clusters. The second step called "inflation", squares each element of the matrix individually, and then renormalizes; this step corresponds to increasing the strength of intra-community ties. This process converges to a stationary matrix with several connected components, which the algorithm output as the communities.

10. **Metis:** Metis [13] is a graph partitioning method which is a variation of the Kernighan-Lin algorithm [14]. Metis partitions a node-weighted network into a specified number of equal weight sets while minimizing the number of edges between the sets. Here we used a version of Metis we adapted for finding high-conductance sets.

In the process of generating examples, we discard communities of size less than 10 or greater than 1000, as well as those communities that contain multiple components. The number of examples extracted varies among the procedures. However, the methods we use for class separability are sensitive to class imbalance. Thus, we under-sample the large

#	Feature	Description
1	n	Number of nodes
2	m	Number of edges
3	Diameter	Greatest distance between two nodes by traversing shortest paths
4	Edge Density	Ratio of m to the maximum possible number of edges
5	Conductance	Ratio of m to the sum of the total degrees of the n nodes, including edges to rest of the network
6	Transitivity	Ratio of the number of 3-node cycles (triangles) to the number of 2-hop paths (open triangles)
7	Triangle Density	Ratio of the number of 3-node cycles (triangles) to the number of possible node triples
8-11	Shortest Path	All pairs shortest paths The features are the three quartiles and the maximum.
12-15	Edge Betweenness	For each edge, fraction of all-pairs shortest paths that include that edge The features are the three quartiles and the maximum.
16-20	Node Betweenness	For each node, fraction of all-pairs shortest paths that include that node The features are the minimum, the three quartiles, and the maximum.
21-25	α	For each nonmember on the fringe of the community, number of members that this node is connected to; The features are the minimum, the three quartiles, and the maximum.
26-30	β	For each member, number of other members this node is connected to The features are the minimum, the three quartiles, and the maximum.
31	Treesum	Total number of spanning trees of the community graph, divided by the total number of spanning trees of a K_n-clique (computed using Kirchoff's matrix tree theorem [19])
32-36	Information Centrality	For each node, its Stephenson and Zelen's information centrality index [28] The features are the minimum, the three quartiles, and the maximum.

Table 1: List of features corresponding to measures of the subgraphs that communities induce.

classes and to a lesser extent over-sample small classes to reduce this source of bias.

Our algorithm selection has the purpose of illustrating the applicability of our framework. The approach, however, is not limited to the list we consider. Our method scales to a large number of classes, and a collection of classes should include enough information to reflect the analysis intended. As discussed in the next section, a pair of classes may be highly correlated to each other (e.g., RW0 and RW15). As a result, they may split the predictions in such a way as to obfuscate the interpretation of the outcome. To avoid this pitfall, an inter-class correlation analysis can be employed to assess the independence of the algorithm selection [29].

3.2.2 *Feature Extraction*

In the feature extraction phase, we measure the subgraphs induced by the communities produced in the previous step and those induced by annotated communities. We use a large spectrum of measurements that cover many properties of both the internal link structure and the external interaction of the community with the rest of the network. Each measurement corresponds to a dimension of our feature space. Table 1 lists the features and describes their corresponding measures.

Most of the features can be understood from their table description. The feature Information Centrality, however, deserves further explanation. This measure captures a node's degree of centrality as a function of how fast its information can sequentially reach every other node in the network. For a given node, the information centrality computes a harmonic mean of the amount of "signal" that a node receives from other nodes. A signal between two nodes corresponds to a path between them, which varies according to the "noise", instantiated here as the path length [28].

By measuring the structural properties described in Table 1 for each example of a community derived in the previous phase, we obtain 11 classes of labeled examples in feature space, which constitute the input in our framework.

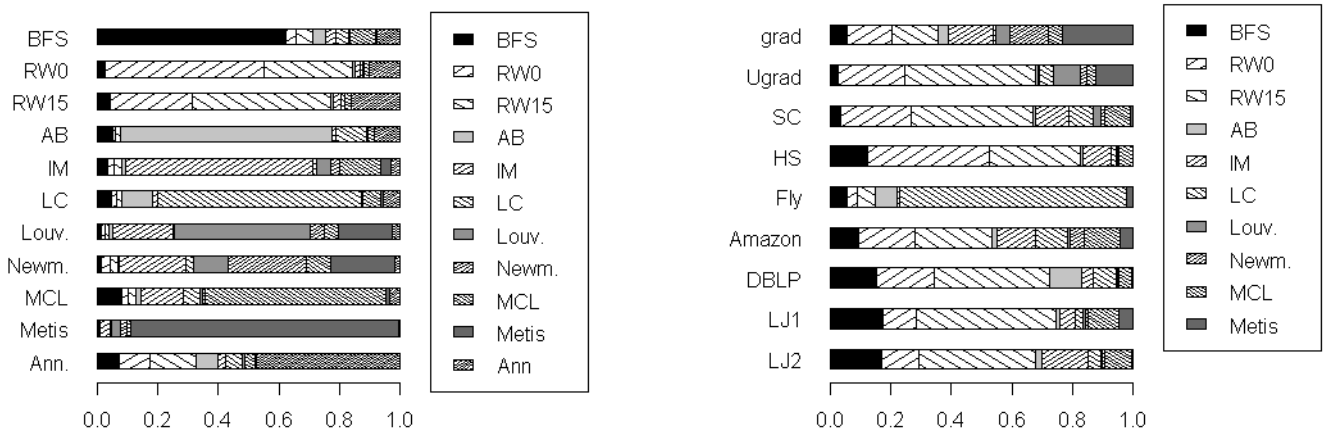
4. FRAMEWORK AND APPLICATION

In this section we present an experimental application using the data we processed through the steps described in the previous section.

4.1 Class Separability Measures

Methods for measuring class separability are popular in machine learning for guiding feature selection analysis. Accordingly, effective feature sets for classification tasks are the ones that simultaneously lead to high inter-class and low intra-class variability [29]. Methods of class separability allow for a rigorous analysis of independence among classes. Unfortunately, many of these methods are computationally demanding or dependent on assumptions which are often mismatched with applications [8].

In this work, we frame the research question of discriminating the structure of different communities as a class separability problem. The separability of structural classes of communities provides information on whether different communities come from the same (or fundamentally different) distributions of feature values. This analysis is informative of the extent to which different algorithms produce structural differences and the extent to which community detection algorithms succeed in producing sets that resemble annotated communities. Aiming at achieving computational scalability and fine-grained separability information,



(a) Class separability via cross validation (DBLP).

(b) Classification of annotated communities.

Figure 1: Distribution of probability mass resulting from the SVM (a) cross validation on the 11 classes, and (b) classification of annotated communities examples and trained on the 10 classes of algorithms.

we use the performance of existing supervised classifiers as a measure of class separability. To our definition, classes are separable to the extent that a classifier can correctly distinguish their structure by exhibiting an accurate classification. More specifically, we employ two techniques, one parametric, namely Support Vector Machines (SVM) [30], and one nonparametric, namely k-Nearest-Neighbors (kNN) [1], to confirm each other's outcomes while ruling out variability due to the specifics of each algorithm. We select hyperparameters in both cases via grid search using the performance of 10-fold cross validation as the objective function. Both methods are capable of handling a large number of classes and scale to a large volume of data.

We measure class separability using the performance of a 3-fold cross-validation. For each network, we train a multiclass classifier on a set containing two thirds of the examples, which are selected at random, and then evaluate the performance of the model on the remaining third, which constitute the test set. We perform 3 rounds of this process and average the outcomes. For each element in a test set, the probabilistic SVM model outputs a probability mass vector indicating the probability that each data point belongs to each class.

4.2 Experimental Analysis

Our primary goal is to gain insight into whether the classes are separable in the feature space defined. Second, building on the preceding observations, we are interested in finding the algorithms whose output structurally reflects the annotated communities. Finally, we study the features that are most useful for distinguishing between the structural classes of network communities.

Our first experiment performs the cross-validation on all the 11 classes. We first observe that the experiments suffer little variability between the two classifiers.

To illustrate the method's output, Figure 1(a) presents the analysis of the outcome produced by the SVM-based method applied to the DBLP network. In the picture, we show a bar graph of the distribution of probability mass for each class derived from the network DBLP. This graph visually demonstrates that the bulk of the probability mass from each class was correctly classified. Table 2 contains a summary of results for all networks. Each entry in the table represents the fraction of probability mass from that class that was correctly assigned. When a value appears in parentheses, this indicates that most of the probability mass was assigned to some other class. While the the SVM provides a breakdown of values by classes, the last two rows present global scores of separability computed using *scatter matrices* [29], which is a standard measure of class separability in pattern recognition[2]. The last row presents a reference point of little separability for each network, where we shuffle the labels of points before computing the score.

As this table shows, only 17 out of 99 network-class pairs failed to have a plurality of the probability mass correctly classified. Newman Modularity is frequently misclassified; however, it is a small class in all networks, especially on the smaller ones (e.g., on network SC, Newman Modularity found only 3 communities of size between 10 and 1000). In the case of annotated communities a plurality of their corresponding classes tends to be correctly classified, with the exception of network Fly. Figure 1(a) serves as a visual reference of a network whose classes have global separability score of 22.7. The other networks exhibit comparable scores, with the exception of network Fly, whose classes are not well separated.

The previous experiment shows that annotated communities tend to form their own, separable class that is significantly distinct from all other classes. However, a question

[2]We use a criterion referred to as J_3 by [29].

	Grad	Ugrad	HS	SC	Fly	DBLP	Amaz	LJ1	LJ2
BFS	60%	88%	73%	70%	(40%)	63%	55%	86%	81%
RW0	44%	55%	43%	(39%)	(27%)	52%	43%	61%	63%
RW15	40%	(29%)	44%	42%	34%	46%	39%	57%	57%
AB	83%	91%	90%	71%	60%	70%	74%	90%	89%
IM	27%	(23%)	72%	73%	(2%)	62%	51%	82%	66%
LC	68%	96%	83%	85%	83%	67%	56%	90%	89%
Louv.	24%	(3%)	49%	(1%)	(0%)	45%	58%	38%	49%
Newm.	(14%)	(25%)	(15%)	(0%)	90%	26%	39%	45%	56%
MCL	19%	(22%)	57%	28%	(34%)	59%	46%	80%	74%
Metis	61%	73%	81%	90%	(42%)	88%	66%	92%	86%
Annot.	37%	33%	50%	46%	(8%)	47%	40%	72%	71%
Global	**19.2**	**22.3**	**26.0**	**27.4**	**6.3**	**22.7**	**16.4**	**19.7**	**21.9**
Ref.	14.7	13.1	13.0	13.0	12.9	13.0	12.9	12.9	12.9

Table 2: Percentage of the probability mass of classification of elements in the test set into the correct class, using SVM, for all networks. The last two rows present global separability scores using scatter matrices.

Network		**Grad**	**Ugrad**	**HS**	**SC**	**Fly**	**DBLP**	**Amaz**	**LJ1**	**LJ2**
Number of Features Selected		**6**	**7**	**10**	**5**	**6**	**10**	**8**	**12**	**11**
Rank	**Feature**									
1	**Conductance**	1	1	1	1		1	1	1	1
1	**Diameter**		1	1	1	1	1	1	1	1
3	**Info Centrality***	2	2	3	1	1	2	1	2	2
4	**Node Betweenness***			2	2		2	1	5	5
5	**Shortest Path***			1		3	2	1	1	1
6	β^*	1	1	1			2	1	1	1
7	α^*	1		1		1		2	1	
Other features**		#6	#4, #7							

Table 3: Summary of the feature selection results. Features are ranked in order of their frequency in the selection list over the networks. (∗ reporting how many quartiles of the property were selected. ∗∗ feature number according to Table 1.)

of interest to the design and application of community detection procedures is what algorithms output communities bearing the closest structural resemblance to the annotated communities. To answer this question we perform a variation of the classification task previously described. We train a classifier on the 10 classes corresponding to the community detection algorithms and leave the class of annotated communities out of the training set. The goal of this experiment is to evaluate to which class of intrinsically defined communities the annotated examples of the test set are classified.

Figure 1(b) shows the distribution of probability mass of the annotated communities classified into the different classes corresponding to community detection algorithms. The structure that random-walk-based algorithms produce is clearly the most similar to that of the annotated communities. For 7 of the 9 networks, a plurality of the probability mass from the annotated communities was assigned to the classes RW15 and RW0. Due to their high similarity, the classifier confuses these two random-walk-based algorithms as shown in Figure 1(a). The exceptions to this are networks Grad and Fly. For Grad, their annotated communities' probability is spread across all classes, where Metis received the plurality of the mass. In the Fly network, the greatest share of the mass of annotated communities is assigned to LC. These exceptions are associated with small network datasets, therefore the variability could be due to small population sample size. Given the diverse nature of these networks, it is perhaps surprising that in virtually all domains the random-walk communities bear the closest structural resemblance to the annotated communities.

4.2.1 *Feature Selection*

As we have seen in the preceding experiment, each community detection algorithm extracts a distinct structure, which our method is able to separate when projected onto the feature space we define. In this section, we are concerned with identifying the ways in which the algorithms produce bias by finding which properties exhibit the highest degree of between-class variability.

To address this question we use the Correlation-based Feature Selection algorithm (CFS) [11] to identify subsets of the most discriminative features for each network. CFS is intended to give a high score to sets of features that are highly predictive of the class and are not redundant with one another. CFS begins with no nodes in the feature set, and it then employs a hill-climbing algorithm to search the space of feature subsets.

Table 3 lists the features selected by CFS for each net-

	Grad	Ugrad	HS	SC	Fly	DBLP	Amaz	LJ1	LJ2
All Features	62.9%	86%	82.2%	80.9%	93.6%	81.3%	65.3%	89.1%	88.5%
With selection	61.5%	84.7%	85.1%	81%	90.6%	79.4%	63%	78.8%	76%

Table 4: k-Nearest-Neighbors classification performance using both the full set of features and the subset of the most discriminative features selected by CFS.

work ranked in order of the frequency with which they appear in the selection over the networks. The table lists the most frequent features, or sets of features for those properties calculated with quartiles. The entries for row "Features" and column "Network" that contain the value 1 indicate the presence of that feature in the feature selection applied to that particular network data, whereas empty cells indicate the absence thereof. Integers larger than 1 can be found in some of the entries and indicate the number of quartiles from that feature that were selected by CFS. In nearly every network, conductance, diameter, information centrality, and node betweeness were the most discriminative features.

Surprisingly, in several cases, multiple quartiles of a feature appear: e.g., Fly has 3 path length quartiles, and LJ1 and LJ2 each contain all 5 node betweenness features. We had expected that different quartiles of the same feature would be highly correlated to each other, and therefore they would be unlikely to co-occur among the features selected by CFS. Instead, these results suggest that varying the choice of community detection algorithm results in fine-grained variation in the distribution of such features.

To assess the effectiveness of the features CFS found, Table 4 presents for all networks the classification performance of the kNN cross validation using both the full set of features and the subset of features found by CFS. We see that in most cases, there is very little loss in accuracy. We observe a similar qualitative outcome for the SVM cross validation. In the table, the largest drops happened for LJ1 and LJ2 and reduced the accuracy by less than 15%. Being nearly as discriminative as the full set, a reduced set containing a handful of features retains the relevant information needed to analyze the bias produced by different algorithms.

We use the sets of the most discriminative features to study which tendencies in feature values are associated with which algorithms. To this end, we conducted a range analysis which distinguishes the different algorithms according to the value of their features. In the interest of space, we summarize the qualitative outcome of this experiment in Table 5. The entries correspond to the bias produced by each of the algorithms, considering all networks. Features take on a varying range of values across different networks. Thus, to label the magnitude of features, we compute the mean value of each class and compute a global median of these averages over all classes. The averages occurring between the 33rd and 67th percentile constitute the *medium* denomination; whereas those below the 33rd and above the 67th constitute *low* and *high*, respectively. Finally, we count how many times each feature produced each of the denominations across all the networks. From this count, we determine the most frequent tendency which make up the entries we present in the table.

Using this analysis, we are able to group algorithms with similar behavior. For example, the random-walk-based algorithms produce the same structural bias. The same holds for

Algorithm	Conduct.	Diam.	Nd.Betw.	Inf.Cent.
Annotated	Medium	High/Low	High	Low
RW0, RW15	Low	High	High	Low
Louvain	High	High	Low	Medium
Newman	High	High	Low	Medium
AB	Medium	Low	Medium	High
LC	Medium	Low	Medium	High
Metis	High	Medium	Medium	Medium
IM	High	Medium	High	Medium

Table 5: Tendency of different algorithms with respect to the most discriminative features.

Louvain and Newman; and AB and LC. The profile of annotated communities is close to that of random-walk-based algorithms, with a few nuances. Annotated communities exhibit medium conductance whereas RW0 and RW15 extract low conductance sets. In addition, the diameter of annotated communities was measured as high for four of the networks, medium for one of them, and low for the remaining four. This contrasts with RW0 and RW15, which produce set with high diameter. Nevertheless, the similarity due to other features explains the ways in which annotated communities resemble the output of random-walk-based algorithms. Finally, Metis and IM differ only in behavior of the node betweenness feature.

5. DISCUSSION

In this paper we tackle the complexity involved in the task of extracting communities in networks by illuminating structural properties of different algorithms and communities that arise in networks across a diverse set of domains. Our approach differs fundamentally from previous work in the area due to its supervised nature. The existing community detection algorithms treat the problem as an unsupervised decomposition of a network with little sensitivity to different purposes, structures of interest, and the various domains of application. Accordingly, our supervised approach may be used by a practitioner to make an informed decision about the most suitable algorithm for a given network in the following way. First, we produce a test set comprising examples of the communities we are interested in finding, which could be either real or synthetic. Second, we choose a set of algorithms we want to evaluate. Finally, we apply our approach using the target network and present the classifier with the test set. The classifier assigns the probability mass of the test set to the class of algorithm that bears close resemblance to the examples. The algorithm that receives the bulk of the mass is the algorithm that may succeed in extracting communities that structurally resemble the ones we are interested in. Researchers may also benefit from our methodology when designing new community detection al-

gorithms as a way to compare the behavior of new methods with existing ones.

Structural similarity is a weaker requirement than accuracy. In other words, communities with similar properties to real communities may not correspond exactly to the communities we may expect to find. Nevertheless, mastering structure is a fundamental stepping stone in the development of algorithms to accurately find the communities of interest.

Finally, our approach suggests a change in the way we approach the problem of community detection. Instead of developing multiple general purpose algorithms that find a particular type of community, one could use a supervised approach that allows the user to specify what they intend to find through examples. Then, one could develop an algorithm that learns from these examples and retrieves similar structures. This is part of our agenda for future work.

Acknowledgments

We are grateful to Eduardo Valle for valuable discussions. Support for this research is provided by AFOSR grants FA9550-09-1-0100 and FA9550-09-1-0675.

6. REFERENCES

[1] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, January 1991.

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

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2006.

[4] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Mar. 2008. Journal of Statistical Mechanics.

[5] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, Dec. 1996.

[6] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111+, Dec. 2004.

[7] S. V. Dongen. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008.

[8] N. Fatemi-Ghomi, P. Palmer, and M. Petrou. The two-point correlation function: A measure of interclass separability. *Journal of Mathematical Imaging and Vision*, 10:7–25, 1999. 10.1023/A:1008362414568.

[9] S. Fortunato. Community detection in graphs. *0906.0612*, June 2010. Phys. Reports 486, 75-174.

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

[11] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, Department of Computer Science, University of Waikato, 1999.

[12] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439, 2006.

[13] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20:359–392, December 1998.

[14] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(1):291–307, 1970.

[15] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80:056117, Nov 2009.

[16] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, 2006.

[17] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. In *Proc. of the 17th Intl. Conf. on World Wide Web*, 2008.

[18] J. Leskovec, K. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th Intl. Conference on World Wide Web*, 2010.

[19] R. Lyons and Y. Peres. *Probability on Trees and Networks*. Cambridge University Press, 2012.

[20] N. Mishra, R. Schreiber, I. Stanton, and R. Tarjan. Finding strongly knit clusters in social networks. *Internet Mathematics*, 5(1):155–174, Jan. 2008.

[21] A. Mislove, B. Viswanath, K. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *Proc. 3rd ACM Intl. Conf. on Web Search and Data Mining*, 2010.

[22] M. Newman. Detecting community structure in networks. *The European Phys. Journal B*, 38(2):321–330–330, Mar. 2004.

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

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

[25] D. Park, R. Singh, M. Baym, C.-S. Liao, and B. Berger. IsoBase: a database of functionally related proteins across PPI networks. *Nucleic Acids Research*, (suppl 1):D295–D300.

[26] P. Pons and M. Latapy. Computing communities in large networks using random walks. *J. of Graph Algorithms and Applications*, 10(2):191–218, 2006.

[27] M. Rosvall and C. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS ONE*, 6(4):e18209, 04 2011.

[28] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37, Mar. 1989.

[29] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 4th edition, Nov. 2008.

[30] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1st edition, Sept. 1998.

[31] E. Weinan, T. Li, and E. Vanden-Eijnden. Optimal partition and effective dynamics of complex networks. *Proceedings of the National Academy of Sciences*, 105(23):7907–7912, June 2008.