Top Leaders Community Detection Approach in Information Networks

Reihaneh Rabbany Khorasgani, Jiyang Chen, Osmar R. Zaïane
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
{rabbanyk, jiyang, zaiane}@cs.ualberta.ca

ABSTRACT

Much of the data of scientific interest, particularly when independence of data is not assumed, can be represented in the form of information networks where data nodes are joined together to form edges corresponding to some kind of associations or relationships. Such information networks abound, like protein interactions in biology, web page hyperlink connections in information retrieval on the Web, cellphone call graphs in telecommunication, co-authorships in bibliometrics, crime event connections in criminology, etc. All these networks, also known as social networks, share a common property, the formation of connected groups of information nodes, called community structures. These groups are densely connected nodes with sparse connections outside the group. Finding these communities is an important task for the discovery of underlying structures in social networks, and has recently attracted much attention in data mining research.

In this paper, we present Top Leaders, a new community mining approach that, simply put, regards a community as a set of followers congregating around a potential leader. Our algorithm starts by identifying promising leaders in a given network then iteratively assembles followers to their closest leaders to form communities, and subsequently finds new leaders in each group around which to gather followers again until convergence. Our intuitions are based on proven observations in social networks and the results are very promising. Experimental results on benchmark networks verify the feasibility and effectiveness of our new community mining approach.

1. INTRODUCTION

Previous data mining approaches, such as association rule mining, supervised classification or clustering algorithms, usually attempt to discover patterns in a data set characterized by a collection of independent instances, which is consistent with the classical statistical inference problem of trying to identify a model given an independent, identically distributed (IID) sample [15]. However, a new emerging challenge for data mining researchers is to solve the pattern discovery problem on data sets which are richly structured and mostly heterogeneous. Such data sets are usually modeled as information networks and contain different object types, which can be related to each other in multiple ways, e.g., commercial data describing connections between customers, products and transactions. Naïvely applying traditional statistical inference procedures, which assume that instances are independent, can lead to inappropriate conclusions about the data [19]. For example, for a search engine, indexing and clustering web pages based on the text content without considering their linking structure would definitely lead to unsatisfactory results for queries. Therefore, for a pattern mining problem, the relations between objects should be taken into consideration and can be very important for understanding the intrinsic structure of the data and knowledge patterns.

Nowadays, in myriad application domains we are collecting inter-related data in the form of networks such as in marketing, biology, epidemiology, sociology, criminology, zoology, etc. A common trait of today's social networks or information networks, and many other scientific datasets, is their community structure, which denotes the existence of densely connected groups of nodes, with sparser connections between these groups. Finding these communities could be of significant practical importance to understand the corresponding data, such as organizational structures, academic collaborations and the user communities in a telecommunication network. For example, groups of web pages that link to more web pages in the community than to pages outside might correspond to sets of web pages on related topics, which can enable search engines and portals to increase the precision and recall of search results by focusing on narrow but topically related subsets of the web [14]. Therefore, *Community Mining*, which focuses on the detection and characterization of such network structure, has received considerable attention over the past few years in social sciences, such as psychology, anthropology, criminology, etc., and lately computer science, particularly data mining.

There are several definitions for communities in the network, but there is no generalized consensus. For instance, a community can be seen as a subgraph such that the density of edges within the subgraph is greater than the density of edges between its nodes and nodes outside it [17]. From that perspective, identifying communities can be seen as finding node clusters in a graph, or graph partitioning. Others have defined a community membership based on a notion of

KDD'10 July 2010, Washington, DC, USA.

structural similarity[39] allowing the additional distinction of hubs that bridge between communities and outliers that are marginally connected.

Recently, many community mining approaches have been proposed to detect communities for various types of social networks [16, 9, 28, 8, 32, 39, 5, 6]. While many show promising results, most of them still present issues. Either they have parameters notoriously difficult to tune or they simply assume that we have no prior knowledge of the communities and the network in question, i.e., there is no further information of the community detection request aside from the network itself. However, this is not always the case. For example, a business network analyst may only show interests in the top-K trading company communities in size; a blog network about US political elections has exactly two communities. Moreover, a visualization systems could help in collecting rough information about a network which could in turn be exploited to discover more accurate structures. Therefore, while proposed methods undeniably have their merit, it is necessary to investigate means for community mining that could utilize any available knowledge of the network to improve the accuracy and efficiency of community detection.

In this paper, we present a new community detection approach based on finding top-k leaders in an information network. Different from previous methods, we exploit prior knowledge of the given network, such as the desired number of communities to be found or the extent of the noise in the data. In our work, we consider that each community has a representative leader node, which is the most central node in the corresponding community, and a community is a set of follower nodes assembling close to a leader. Briefly, our approach first finds promising leader nodes in the given network, then iteratively updates communities and their corresponding leaders until there is no change in the communities. This scheme is very similar to the partitioning philosophy adopted in clustering such as with k-means, except that while k-means is infamously sensitive to noise, our approach, Top-Leaders, allows us to identify marginal nodes in a network as outliers and thus is not affected by noise.

After discussing related work in Section 2, we propose our Top-Leaders algorithm in Section 3. We show later in the experiments in Section 4 that our approach accurately extracts communities based on our iterative strategy. In comparison with other known methods, we consistently discover the most accurate results.

2. RELATED WORK

The research topic of social network analysis is not new and has been the focus of many scholars in anthropology and psycology for many decades. Detecting communities in a social network structure has also been pursued by sociologists and more recently physisists and applied matematiciens with applications to social and biological networks [26]. With the availability of large datasets of information networks, computer scientists have also joined the effort and community mining is becoming a very popular research endeavour. This line of research is addressing similar question as graph partitioning. There are, however, important differences between the goal and applications of the two camps, graph partitioning and community mining, that make quite different technical approaches desirable [27].

There is a long tradition of research by computer scientists on graph partitioning, which is described as dividing the vertices of a network into some number of k groups with roughly equal size, while minimizing the number of edges that run between vertices in different groups. Generally, finding an exact solution for a partitioning task is believed to be an NP-complete problem, making it extremely difficult to solve for large graphs. Nevertheless, a wide variety of algorithms have been developed that provide high quality solutions in many cases: METIS [20], flow-based methods [14], information-theoretic methods [11] and the Kernighan-Lin algorithm [21]. The major incompatibility of these methods is that *community structure detection* assumes that the networks divide *naturally* into some partitions and there is no reason that these partitions should be of the same size.

Another important family of graph partitioning algorithms is the spectral clustering method [29], which divides the network into two groups by looking at the eigenvector corresponding to the second lowest eigenvalue of the adjacency graph Laplacian and separating the vertices by whether the element is greater than or less than zero. Division into a larger number of groups is usually achieved by repeated bisection. Unfortunately, the sizes of the groups into which the network is divided need to be fixed, but are usually unknown beforehand. Additionally, if we set the group sizes to be unconstrained, the method (and other methods that minimize cut size without constraints on the group size) could break down: the minimum cut size is always achieved by the trivial division which puts all vertices in one group and none in the others. Several methods have been proposed to fix the problem, such as *ratio cut* [4], *normalized cut* [36], and the *min-max cut* [12]. However, these approaches are biased in favor of divisions into equal-sized parts.

Therefore, in their study of social networks, sociologist have taken no notice of the aforementioned spectral clustering or other graph partitioning methods, and have instead adopted hierarchical clustering [35] as the standard method.

The main idea of the hierarchical clustering method is to discover natural divisions of social networks into groups, based on a metric measuring the similarity between vertices in a graph. There are many similarity metrics proposed to measure the community relation in networks [5, 9, 16, 18, 28, 30]. Among them, the Modularity Q [28, 9], is the most effective in finding communities [10] and serves as the basis of many other later proposed metrics.

The modularity Q is proposed as a measure of the quality of a particular division of a network. The basic idea is to compare the division to a randomized network with exactly the same vertices and same degree, in which edges are placed randomly without regard to community structure.

Values of Q that are close to 1, which is the maximum, indicates strong community structure. Q typically falls in the range from 0.3 to 0.7 [16] and high values are rare.

In addition to the prominent Q-modularity approach [9] that we mention above and against which we compare our results, there are two other worth mentioning algorithms that are not only innovative in the process of discovering communities but are also highly effective in many cases: CFinder published in Nature and SCAN published in KDD. We also compare our results with these two contenders. Palla et al. [32] propose the CFinder system to partition complex networks into overlapping communities. Based on the observation that typical communities consist of several complete subgraphs sharing many of their nodes, the authors define a community as the union of complete subgraphs of size k. In other words a k-clique community is the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques, where "adjacent" means two k-cliques sharing $k-1$ nodes and k is a given parameter as the clique size. There are some parts of the whole network that are not reachable from a particular k-clique, but they potentially contain their own k-clique communities. Therefore, a single node can belong to several communities. Gathering adjacent small k-cliques to form communities in large networks is very efficient but the authors also showed that with k between 3 and 5 the approach is very effective on real world information networks.

Rooted in the well known density-based clustering algorithm DBScan [13], Xu et al. derived a similar approach for information networks and proposed the SCAN algorithm [39] to detect not only clusters, but also hubs and outliers in networks. Like the notion of reachability in DBScan, the neighbourhood of a vertex is used as a criteria for clustering and nodes that are structurally reachable from each other are grouped together in the same community. Their performance appears very good but it is highly dependant on the two parameters: the structural similarity threshold for a "core" vertex and the minimum number of neighbours needed to propagate the reachability. This sensitivity of the parameters was addressed using a visual data mining approach [7]. The latter approach is also useful as a means to get a sense of the number of the natural communities that exist in a network and can be used as preprocessing for our own algorithm to identify the the number of leaders k if it is not apriori known. Moreover, the FastModularity [9] based on the Q-Modularity can also be used as precursor to our algorithm to determine k before improving on the community membership accuracy with our approach.

3. TOP LEADERS APPROACH

Top-Leaders consists of selecting k representative nodes, that we call leaders, then associate remaining nodes of the network, the followers, to leaders to form communities. An iterative process till convergence elects new leaders for each community and reassigns nodes to the leaders to form new communities. Convergence is attained when the best leaders are found and each node is associated to its most appropriate leader. In this section we present in detail this new approach for detecting communities and elucidate the processes of selecting the initial leaders, associating followers to a leader, and electing new leaders. The basic idea of our approach is inspired by the well-known k-means clustering algorithm. However, there are many differences, particularly the fact that we can detect outliers. Similar to k-means, our algorithm is sensitive to its initialization, the selection of the initial k leaders. We have experimented with a variety of strategies, from a random selection à la k-means to more advanced heuristics. We present in this section some of these strategies.

3.1 Common Framework

A community leader is the most central member in a community and each community is constituted of a leader and the follower nodes associated to it. Some nodes in the network may not be associated to any given leader and thus do not belong to any community and are considered outliers. The basic idea of the Top Leaders algorithm, is to first find some k community leaders, and then determine the community membership of other nodes in the network based on their relations to the identified leaders.

Algorithm 1 highlights the major steps of Top Leaders. The first step is the selection of the initial k leaders by some heuristics. This initialization is described below in Section 3.3. The second step is an iteration in which we alternate between association of followers and election of new leaders. First, nodes are either associated to a leader or labeled as outliers (elaborated further in Algorithm 2), and second, when all nodes in the network are dealt with, a new leader is picked in each community. The reassignment of leaders is simply the election of the node with the highest centrality in a community.

$$\underset{n \in Community(l)}{\arg\max} \quad Centrality(n)$$

The centrality of nodes in a community measures the relative importance of a node within that group. There are many measures of centrality that could be parameter to the algorithm, namely degree, betweenness, closeness and eigenvector centrality measures. We experimented with them all and based on our results, we selected the degree centrality for the default measure which yeilds the most accurate results in most of the cases and also is easy to compute. The degree centrality for a node n within a community is simply the number of edges from the community incident upon n and represents to some extent the "popularity" of n in the community. For a community C of size N, the degree centrality of a node n in C is

$$DC(n) = \frac{deg(n, C)}{N-1}$$

where $deg(n, C)$ is the number of edges in C incident upon n. Consequently, for each community, the centrality of every member is computed and the node with the highest degree is selected as new leader.

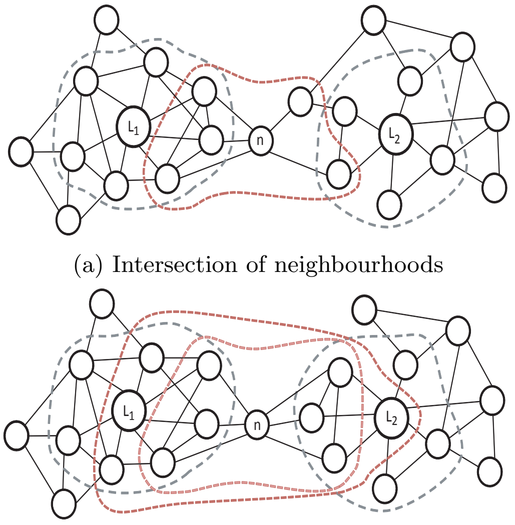
Algorithm 1 Top Leaders algorithm

```
Input: A social network G, and k the number of desired communities
  initialize k leaders
  repeat
    {finding communities}
    for all Node n ∈ G do
      if n ∉ leaders then
        associate n to a leader {Algorithm 2}
      end if
    end for
    {updating leaders}
    for all l ∈ leaders do
      l ← $\arg\max_{n \in Community(l)} Centrality(n)$
    end for
  until there is no change in the leaders
```

Below, we focus on the initialization of leaders and on how to associate nodes to leaders.

3.2 Associating Nodes to Leaders

With leaders representing communities, the community membership of the remaining nodes is the association of followers to nearby leaders. For each node, n, in the network its relation to each leader, l, is assessed by computing the



(a) Intersection of neighbourhoods

(b) Expanding Neighbourhoods

Figure 1: Determining community of node n: n should be assigned to leader L_1 because **a)** n has more common neighbours with L_1 than L_2, **b)** although n has the same number of common neighbours with L_1 and L_2, it has more common neighbours with L_1 if we expand its neighbourhood boundary by one.

size of intersection of this leader's neighbourhood and the n's neighbourhood, i.e., how many neighbours they have in common (Figure 1a).

Moreover, for finding the leader for a given node, we consider different levels of neighbourhoods for that node. We start assessment by neighbourhood depth 1 (which consists of the nodes that are directly connected to this node). If there are more than one leader for this node we would expand the node's neighbourhood by one (by adding nodes that are directly connected to the current nodes in its neighbourhood). We keep expanding the neighbourhoods as long as there are ties up to the neighbourhood depth threshold(δ).

Algorithm 2 depicts the process of associating a node to its appropriate leader. Where $\aleph(n, d)$ denote the set of nodes in neighbourhood depth d of node n. And $|S|$ shows cardinality(size) of set S.

To detect outliers in the network, we define an outlier threshold (γ). Only leaders that have more common nodes than this threshold with the given node are considered. If after reaching the neighborhood threshold, the node is still not assigned to any of the current leaders; it is marked as an outlier. Hubs are those nodes that follow more than one leader. They sit on the intersection of communities. The algorithm is not sensitive to the value of δ. We have experimented with a variety of networks and different depth have given the same result. Thus, we set the δ threshold at 2. The γ outlier threshold can on the other hand give different results. When we know that no outliers exist in the network, γ is set to 0. Otherwise γ depends on the density of the network, and to correctly identify outliers, γ could vary between 1 and 4 in most cases.

3.3 Initialization Methods

Like with any partitioning process, the initialization is crucial. Starting with the correct leaders allows quick convergence while starting with the wrong leaders will necessitate many iterations and may get stuck in a bad local optimum. This is a known problem with k-means for instance and many ways out were suggested in the literature such as running the nondeterministic algorithm multiple times or suggesting heuristics for the selection of better starting points. We experimented with myriad strategies specific to information networks and report here some initializations from a naïve random selection of leaders to a more elaborate approach. We highlight herein these initialization methods and show their impact in the experiments in Section 4.

Algorithm 2 Associate n to its leader

```
Input: Social network G, node n, set of k leaders

  depth ← 1
  CanList ← leaders
  repeat
      CanList ← $\underset{\substack{c \in CandList \wedge \\ |\aleph(n_1,d)\cap\aleph(n_2,d)|>\gamma}}{\arg\max} |\aleph(n_1,d)\cap\aleph(n_2,d)|$
    depth ← depth+1
  until |CanList| ≤ 1 ∨ depth > $\delta$

  if |CanList| = 0 then {No candidate leader}
    associate n as an outlier
  else if |CanList| > 1 then {Many candidates}
    associate n as a hub
  else {Only one candidate leader in CanList}
    associate n to CanList
  end if
```

3.3.1 *Naïve Initialization*

The naïve initialization is a random selection of k nodes from the network. This is simple but is not deterministic and may lead to bad results.

3.3.2 *Top Global Leaders*

Founded on the fact that community leaders are central nodes in their community, this initialization method picks the k most central nodes in the network as the initial leaders. This approach is also naïve because while community leaders are indeed central in their respective communities there is no reason that they should be the most central in the global network. In fact this methods produces good results in some cases but it yielded results worse than random in some others. On average, however, this strategy seems satisfactory. We added a variation such as selecting randomly k leaders from a larger set ck of most central nodes in the graph where c is a constant. However, this did not produce better results and in addition is non deterministic.

3.3.3 *Top Leaders & not Direct Neighbour*

The major drawback of Top-Global-Leaders is the fact that it is possible that two of the most central nodes belong to the same community. Choosing arbitrarily the k most central nodes may force a community to split and this would negatively affect the final results. Therefore we propose to choose the k most central nodes that are not directly connected to each other which avoids choosing leaders in the same community. To implement this strategy we start from the most central node, and add the next central one to the current set of leaders if it is not directly connected to any of the already selected leaders.

3.3.4 Top leaders & Few Neighbours in Common

Top Leaders & not Direct Neighbour method improves the result significantly but still produces inaccurate results in some cases. This intermittent inaccuracy is due to the fact that being direct neighbours does not exclude being in different communities. Therefore, the method could occasionally mistakenly avoid choosing two correct leaders that are directly connected but truly in different communities. To steer clear from this problem, instead of using a simple direct connectivity, similar to the main framework, we assess the intersections to check if these nodes belong to the same community. The computation is simply after starting with the most central node, we add the next central one to the current set of leaders after checking its intersection size to the already selected leaders. In our implementation, two leaders would be deemed to have less than 5 common neighbours. This value was tested and is stable with most networks we encountered.

4. EXPERIMENTS

We evaluated the accuracy of our proposed approach by comparing its results with the true communities in well-known data sets for which we have ground truth. Here we introduce these data sets and evaluation metrics, then report and discuss our results on these data sets.

4.1 Data sets

4.1.1 Karate Club

This network is drawn from the well-known “karate club” study of Zachary [40]. In this study, relations between 34 members of a karate club over a period of two years are observed. During the study, a disagreement developed between the administrator and the teacher of the club, which eventually made the club split into two smaller ones, centering around the administrator and the teacher, represented by node 34 and node 1. Zachary was able to construct a network of friendships, using a variety of measures to estimate the strength of ties between members.

4.1.2 Strike

This is the communication network of employees in a sawmill [31]. This data is collected in order to analyze the communication structure among the employees after a strike. An edge in the network means that the two connected employees have discussed the strike with each other very often. There are three groups according to age and language.

4.1.3 Football

This dataset is the schedule for 787 games of the 2006 National Collegiate Athletic Association (NCAA) Football Bowl Subdivision [39]. In the NCAA network, there are 115 universities divided into 11 conferences. Additionally, there are 4 independent schools, namely Navy, Army, Notre Dame and Temple, as well as 61 schools from lower divisions. Each school in a conference plays more often with schools in the same conference than schools outside. Independent schools do not belong to any conference and play with teams in all conferences, while lower division teams play very few games. The network contains 180 vertices (115 nodes as 11 communities, 4 hubs and 61 outliers), connected by 787 edges.

4.2 Evaluation Metrics

We evaluated extracted communities by both comparing with ground truth and by measuring their modularity.

4.2.1 Comparing with Ground Truth

With ground truth validation is simply accomplished by means of comparison of communities, those discovered against the known communities. We used measures of agreement between partitions typically employed for clustering evaluations: purity and Adjusted Rand Index. Let V be the set of all nodes in communities at let $R = R_1, R_2, \ldots, R_k$ be a partitioning on V such that $V = \bigcup_1^k R_i$ and $R_i \cap R_j = \phi$ for all $i \neq j$. Let n be the size of V ($n = |V|$), and assume R and G are two different partitioning of the same data.

Purity is the number of correctly assigned nodes divided by the total number of nodes in V. Purity ranges from 0 (no agreement at all) to 1 (full agreement). It is computed using the following formula [24]:

$$purity(R, G) = \frac{1}{n} \times \sum_j max_i |R_j \cap G_i|$$

Adjusted Rand Index (ARI) The measure penalizes false negatives and false positives. ARI ranges between -1 (no agreement at all) and 1 (full agreement) with expected value of 0 for agreement no better than random. Let a,b,c and d denote the number of pairs of nodes that are respectively in the same community in both G and R, in the same community in G but in different communities in R, in different communities in G but in the same community in R, and in different communities in both G and R. Then the ARI is computed by the following formula [33]:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c) + (c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c) + (c+d)(b+d)]}$$

$$a = \textstyle\sum_{vw} \delta(R^v, R^w)\delta(G^v, G^w)$$
$$b = \textstyle\sum_{vw} (1 - \delta(R^v, R^w))\delta(G^v, G^w)$$
$$\ldots$$

Where $R^n = \{R_i | n \in R_i\}$ and $\delta(R_1, R_2)$ is 1 if $R_1 = R_2$ and 0 otherwise (c and d obtained similarly). To compare data sets with outliers, we simply treat outliers as another community.

4.2.2 Modularity

When ground truth is not available, modularity (Q) is typically used to assess the quality of discovered communities. It measures how well the edges fall within the detected communities compared to a randomized network. More formally, assume R is a partition of the network with adjacency matrix A containing m edges ($m = \frac{1}{2}\sum_{vw} A_{vw}$); then, given $\delta(R_1, R_2)$ is 1 if $R_1 = R_2$ and 0 otherwise, the portion of edges within communities is

$$\frac{1}{2m} \sum_{vw} A_{vw} \delta(R^v, R^w)$$

The modularity is defined as substraction of this quantity from its expected value for a randomized network (with same nodes where edges created randomly but with respect the node degrees). Which can be computed as

$$Q = \frac{1}{2m}\sum_{vw}[A_{vw} - \frac{d_v d_w}{2m}]\delta(R^v, R^w)$$

Where d_v denotes the degree of node v ($d_v = \sum_w A_{vw}$). The modularity would be zero when the portion of within edge communities is no different than what we expect from a randomized network, and a value higher than 0.3 is a sign for significantly good partition [9]. For networks with discovered outliers, modularity is computed without the outliers.

4.3 Results and Discussions

We report the results for different initialization methods and show a comparison with three of other well-known community detection methods.

4.3.1 Initialization methods

Table 1 shows the improvement of our results by developing the initialization of our algorithm. As shown in Table 1, even the Naïve initialization gives reasonable results but with high variance. The Top Global Leaders improves the results significantly and reaches the maximum ARI in Karate and Strike but the best cases in the Naïve initialization for the Football data set still do better. This indicates room for improvement. Examining the initial leaders obtained from the Top Global Leaders (TGL) and locating them in the network, indicates that some of these leaders are in the same community in the ground truth and choosing them as leaders, forces that community to split.

Top Leaders & not Direct Neighbour (TL&NDN) initialization do not improve the results which shows that the condition of not being direct neighbour is not a good one; since it would not avoid choosing leaders in the same community if they are not directly connected which is very probable. It also may avoid choosing two true leaders which are in different communities and directly connected. The former is also very probable as the leaders are nodes with high centrality and may have links to outside of the community.

Top Leaders & Few Neighbours in Common (TL&FNiC) method gives us the best result. This method makes a greedy selection, starting from the node with the highest centrality to the lowest, we chose one if it does not have more than a threshold neighbours in common with any of the current leaders.

The visualized results of these three data sets are presented in Figure 2. These figures also show the correct communities as we obtained ARI 1 except for the football data set where we misidentified four hubs and assigned them to one community.

4.3.2 Parameters

Our main parameter is the number of communities in the network. This should either be given by domain experts or obtained from another algorithm for community mining that does not require k. However, even the algorithms claiming not requiring this parameter do not find the right number in many cases: fastModularity and cFinder find 3 and Scan finds 4 communities in Karate which has really only 2. For Strike with 3 communities, cFinder finds 6. Similarly for Football there are discrepancies (Table 2). Based on results in Table 2, using TopLeaders after another community detection approach provides k, would increase the quality of the final results significantly based on Modularity. In other words, if we trust the number of communities discovered by a community mining approach, seeding this number to TopLeaders would increase the quality of discovered communities.

Table 1: Results of different initialization methods. For the Naïve method, average±standard deviation is calculated over 100 runs. All the results have the same default parameters (neighborhood threshold = 2, initialization threshold = 5 (nodes in common)) except the number of communities for each data set. (karate=2, strike=3, football=11) and the outlier threshold which is 4 for football but zero for karate and strike as we do not want to detect any outliers in those data sets.

method	dataset	ARI	purity	Q
Naïve	Karate	.80±.33	.90±.20	.28±.13
	Strike	.59±.25	.81±.13	.41±.12
	Football	.39±.12	.66±.08	.27±.07
TGL	Karate	1.0	1.0	0.37
	Strike	1.0	1.0	.54
	Football	.83	.88	.43
TL&NDN	Karate	1.0	1.0	0.37
	Strike	1.0	1.0	.54
	Football	.78	.88	.42
TL&FNiC	Karate	1.0	1.0	0.37
	Strike	1.0	1.0	.54
	Football	.98	.97	.51

We have three more parameters which are all integers in small range and easy to tune. The first one is the outlier threshold, γ. This parameter shows how many common nighbours should a node have with a leader to be consider connected to that leader. That is, if a node has less than γ common friends with the leader, it does not have a considerable intersection with it and could not be part of its community. γ is an integer set to 0 (no outlier detection), left at 4 (default value), or adjusted according to desired amount of noise that should be removed.

For the two other parameters, our algorithm does not exhibit any sensitivity and they can remain at their default values. The first is the threshold used in the Top Leaders & not Direct Neighbour initialization. It is set to 5 in all our experiments, indicating that two leaders may be in the same community if they have more than 5 common friends. The second is the neighbourhood threshold, δ, which shows to what extent we should expand our neighborhood to find the winner leader. Setting this parameter to 2 gives us the reported results and increasing it does not change the results.

4.3.3 Comparing with other approaches

Tables 2 shows a comparison between our approach and three well-known algorithm: FastModularity, cFinder and SCAN. Given the correct initial k, TopLeaders always provides the best result. The other methods do not always find the correct k but even when seeded to TopLeaders, our approach improved the quality of the found communities based on ARI.

Table 2: Comparision with other approaches. Column k indicates the number of communities obtained

dataset	method	k	ARI	purity	Q
Karate 2 groups	fastModularity	3	.680	.970	.380
	cFinder	3	.705	.065	.182
	TopLeader(3)		**.838**	1.0	.374
	SCAN	4	.314	.764	.312
	TopLeader(4)		**.788**	1.0	.361
	TopLeader(2)		1.0	1.0	.371
Strike 3 groups	fastModularity	4	.664	.958	.555
	TopLeader(4)		**.935**	1.0	.532
	cFinder	6	.348	1.0	.485
	TopLeader(6)		**.609**	1.0	.457
	SCAN	3	.848	.958	.547
	TopLeader(3)		**1.0**	1.0	0.548
Football 11 groups	fastModularity	7	.206	.427	.567
	TopLeader(7)		**.637**	.783	.394
	cFinder	12	.983	.913	.532
	TopLeader(12)		**.993**	.977	.511
	SCAN	11	1.0	1.0	.501
	TopLeader(11)		.988	.977	.513

5. CONCLUSIONS

We introduced a novel algorithm, Top Leaders, to mine communities in an information network, which assigns nodes to leaders of communities and selects the leaders of communities iteratively. This algorithm is effective in discovering communities and also in identifying outliers in a network. We applied the algorithm to known real world networks with ground truth. The experimental results confirm the accuracy and effectiveness of our algorithm. *Top Leaders* requires k, the number of desired communities as input. This may seem a major hurdle. However, it is possible to obtain k after running other contenders such as FastModularity, SCAN or CFinder and provide the number of discovered communities to our algorithm. Our experimental results proved that communities obtained in this way are more accurate than the original discovered communities even if the used method detected wrong number of communities.

6. REFERENCES

[1] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. En event-based framework for characterizing the evolutionary behavior of interaction graphs. In *ACM SIGKDD)*, pages 913–921, 2007.

[2] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *ACM SIGKDD*, pages 44–54, 2006.

[3] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda. Detection of complex networks modularity by dynamical clustering. *Physical Review E*, 75:045102, 2007.

[4] P.K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *International Conference on Design Automation*, pages 749–754, 1993.

[5] J. Chen, O. R. Zaïane, and R. Goebel. Detecting communities in social networks using max-min modularity. In *SIAM Data Mining Conference*, pages 105–112, 2009.

[6] J. Chen, O. R. Zaïane, and R. Goebel. Local community identification in social networks. In *Proceedings of the International Conference on Advances in Social Network Analysis and Mining(ASONAM)*, pages 237–242, 2009.

[7] J. Chen, O. R. Zaïane, and R. Goebel. A visual data mining approach to find overlapping communities in networks. In *International Conference on Advances in Social Network Analysis and Mining*, pages 338–343, 2009.

[8] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72:026132, 2005.

[9] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[10] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *J. Stat. Mech*, page P09008, 2005.

[11] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *ACM SIGKDD*, pages 89–98, 2003.

[12] Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *IEEE ICDM*, pages 107–114, 2001.

[13] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD)*, pages 226–231, 1996.

[14] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *ACM SIGKDD*, pages 150–160, 2000.

[15] Lise Getoor and Christopher P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations*, 7(2):3–12, 2005.

[16] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. In *PNAS USA, 99:8271-8276*, 2002.

[17] S. Gregory. An algorithm to find overlapping community structure in networks. In *PKDD*, pages 91–102, 2007.

[18] R. Guimera, M. Sales-pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, 2004.

[19] D. Jensen. Statistical challenges to inductive inference in linked data, 1999.

[20] George Karypis and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distriuted Computing*, 48(1):96–129, 1998.

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

[22] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78:046110, 2008.

[23] F. Luo, J. Z. Wang, and E. Promislow. Exploring local community structures in large networks. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 233–239, 2006.

[24] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[25] T. Nepusz, A. Petroczi, L. Negyessy, and F. Bazso. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77, 2008.

[26] M. E. J. Newman. Detecting community structure in networks. *Eur. Phys. J.B*, 38:321–330, 2004.

[27] M. E. J. Newman. Modularity and community structure in networks. *PNAS USA*, 103, 2006.

[28] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 2004.

[29] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[30] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending modularity definition for directed graphs with overlapping communities, Eprint arXiv:0801.1647 at arxiv.org. 2008.

[31] Pajek. http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[32] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814, 2005.

[33] Jorge M. Santos and Mark Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *ICANN (2)*, pages 175–184, 2009.

[34] Purnamrita Sarkar and Andrew Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations: Special Edition on Link Mining*, 7(2):31–40, 2005.

[35] J. Scott. Social network analysis: A handbook, Sage, London 2nd edition(2000).

[36] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 2000.

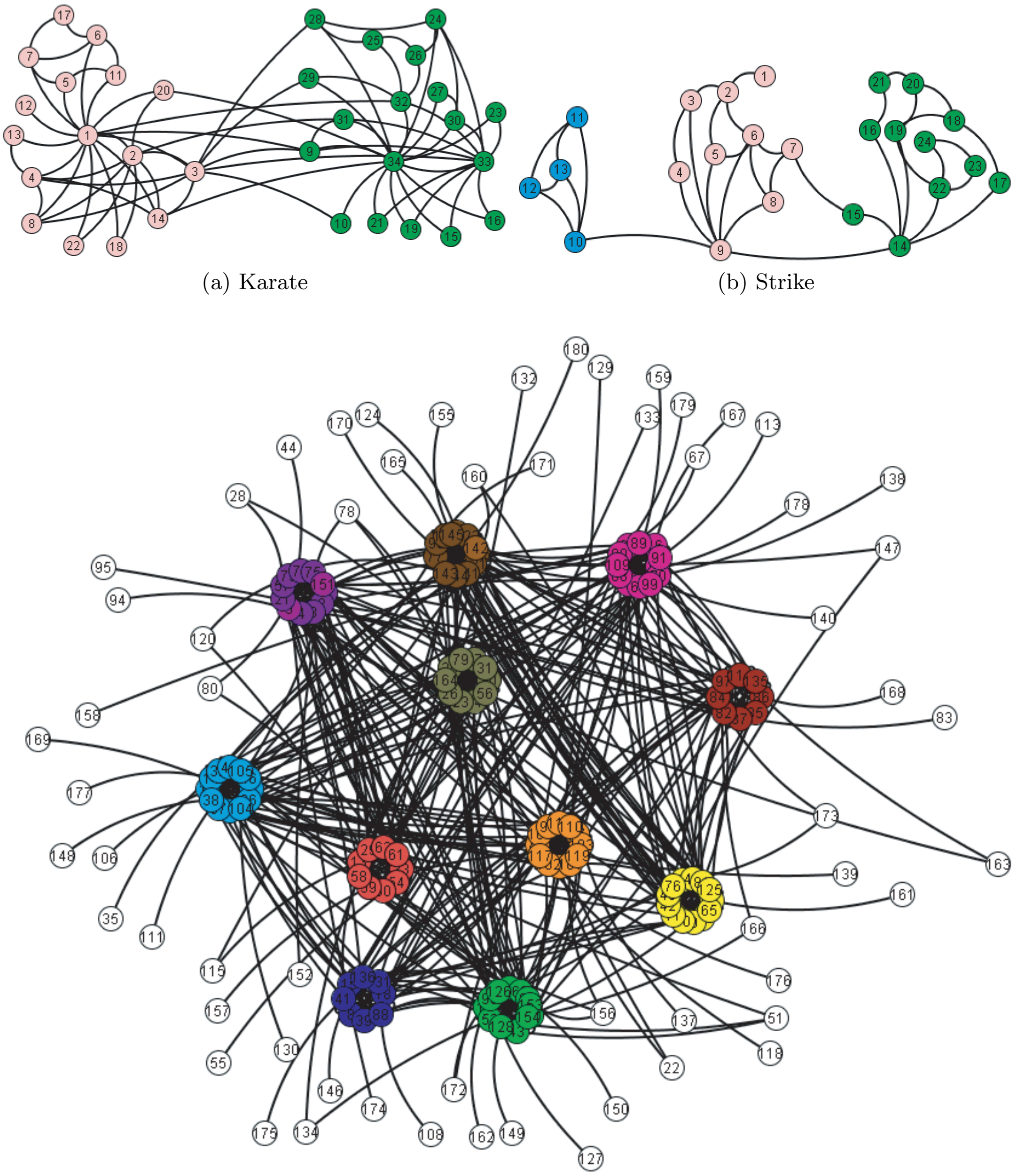
[37] David Strang and Soule Sarah A. Diffusion in organizations and social movements: From hybrid corn to poison pills. *Annual Review of Sociology*, 24(1):265–290, 1998.

[38] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *ACM SIGKDD*, pages 717–726, 2007.

[39] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *ACM SIGKDD*, pages 824–833, 2007.

[40] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[41] S. Zhang, R.-S. Wang, and X.-S. Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A*, 374:483–490, 2007.



(a) Karate

(b) Strike

(c) Football

Figure 2: Visualized communities detected using Top Leaders algorithm. These communities correspond to the results reported on the last row of Table 1.