A Diffusion of Innovation-Based Closeness Measure for Network Associations

Reihaneh Rabbany Khorasgani, Osmar R. Zaïane
Department of Computing Science
University of Alberta
Edmonton, Canada
rabbanyk,zaiane@ualberta.com

***Abstract*—Network association is a prevalent representation when dealing with data from present-day applications. Examples are crime event connections in criminology, cellphone call graphs in telecommunication, co-authorship networks in bibliometrics, etc. A large body of work has been devoted to the analysis of these networks and the discovery of their underlying structures. One important structure is the notion of community *i.e.* a group of nodes that are relatively cohesive within and reasonably disjointed outside. Finding the communities usually relies on a closeness/distance measure between network nodes. In this paper, we propose a novel closeness measure, named iCloseness, inspired by the theory of Diffusion of Innovations in anthropology. It is computed based on the intersection of neighbourhoods and quantifies the closeness of two nodes. To apply this measure we adjusted the Top Leaders community mining method to use this measure for community detection. Experimental results on real world and synthesized information networks show the effectiveness of our proposed measure and highly motivate the application of the iCloseness measure in the context of community mining.**

***Keywords*-Social Network Analysis; Closeness Measure; Community Mining**

I. Introduction

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 data sets, 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. 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 in computer science, particularly in 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 sub-graph such that the density of edges within the sub-graph is greater than the density of edges between its nodes and nodes outside it [1]. 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 structural similarity[2] allowing the additional distinction of hubs that bridge between communities and outliers that are marginally connected.

Alternatively, we could assume communities as groups of nodes in the network that are similar to each other. Where nodes are associated with the community whose members are most similar to it. In typical clustering approaches, data items are usually represented by feature vectors while their similarity is computed by a pattern proximity measure, usually a distance function defined on pairs of data points (*e.g.* Euclidean distance, Mahalanobis distance). There are few similarity/distance measures proposed for relational data in the form of graph or network [3]. However none are commonly used or properly suitable in the context of community mining.

In this paper we proposed a new similarity/closeness method for graphs, which is appropriate for mining communities. This closeness measure, called Intersection Closeness (*iCloseness* for short), assesses the relations between community nodes. ICloseness encapsulates the notion of membership in a community and its basic idea is reinforced by observations made on community dynamics in social networks with regard to the probability of joining a group based on the concept of Diffusion of Innovation [4]. It is observed that the likelihood of joining a community in social networks depends upon the number of pre-existing connections with group members and the density of edges between these members and other members in the group. In other words, if I am faced with two groups in which I already have friends and if I need to join one of these groups, I could choose either one but, there is a higher probability to join the group in which I have more friends. In addition if I am faced with two groups in which I have the same number of friends, there would be a higher probability that I would join the group in which the connectivity of my friends with the group is stronger.

To present applicability of the *iCloseness* in community mining, we adjusted our Top Leaders community detection approach introduced in [5] to use our iCloseness as its proximity measure. *iTop Leaders*, this new version of Top Leaders coupled with our notion of closeness highlighted

herein, exhibits promising results both on real and synthetic data sets. It alos allows us to identify marginal nodes in a network as outliers and thus is not affected by noise. Moreover, hubs, nodes that connect different communities, can also be identified.

After discussing related work in Section II, we detail the iCloseness measure and propose the adjusted iTop Leaders algorithm in Section III. We show later in the experiments in Section IV that the proposed mining approach leveraged with *iCloseness* accurately extracts communities. In comparison with other measures and methods, we consistently discover the most accurate results.

II. Related Work

The research topic of social network analysis is not new and has been the focus of many scholars in anthropology and psychology for many decades. Detecting communities in a social network has also been pursued by sociologists and more recently physicists and applied mathematicians with applications to social and biological networks [6]. With the availability of large real world 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 [7].

One important family of graph partitioning algorithms is the spectral clustering method [8], 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* [9], *normalized cut* [10], and the *min-max cut* [11]. However, these approaches are biased in favour of divisions into equal-sized parts.

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. 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 [12] 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/closeness between vertices and/or a quality function measureing the quality of a particular division. [13], [14], [15], [16], [17], [18]. The modularity Q [17], [14], is a well-known example of such quality function in finding communities [19] and serves as the basis of many other later proposed metrics.

The closeness measure presented in this paper is based on the connection between theoretical models of diffusion in social networks to the community membership investigated by Backstrom *et al.* [4]. They studied the evolution of large-scale social networks over time and found that the tendency of an individual to join a community is influenced by their number of friends in that community and also crucially by how those friends are connected to one another.

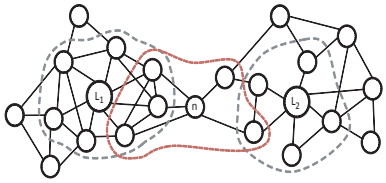
III. Method

Here we first introduce our proposed measure of closeness – *iCloseness*(Section III-A). We then continue in Section III-B by overviewing the Top Leaders Community Mining approach and present our adjustments to this algorithm mainly how it applies *iCloseness* in associating nodes to the communities.

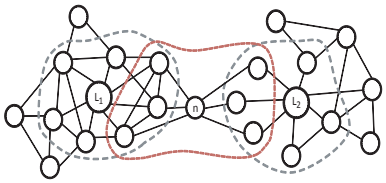
A. iCloseness

Defined based on theory of *Diffusion of Innovations* and its application to information networks [20], [21], *iCloseness* measures closeness between two nodes. It indicates how much these nodes tend to belong to the same community and is calculated based on their common neighbours. Diffusion of Innovations which stems from research in sociology, is a theory of how, why, and at what rate new ideas and technology spread through cultures. Specifically for the case of information networks, if we consider the act of joining a community as a behaviour that spreads within a network the idea of Diffusion of Innovations very naturally extends to community mining. Socially, there is indeed advantage in joining a group that already includes friends that know each other and who are connected. Backstrom *et al.* further discuss this idea in the context of social networks dynamics [4]. Accordingly the probability of joining a community depends on the number of friends one already has in the community and the internal connectedness of the friends within it. The Intersection Closeness (*iCloseness*), which is based on the common neighbours of two nodes within a predefined neighbourhood, is our central mean to reflects this idea.

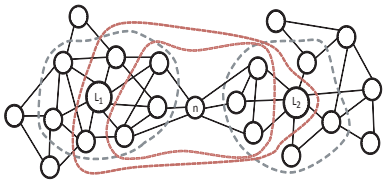
Figure 1, illustrates the idea behind *iCloseness* using an example. Suppose we want to assign node n to either one of the two communities led by L_1 and L_2. The important factors are **a)** Number of n's neighbours in each community (Figure 1(a)). One prefers a group in which one has more friends. **b)** Connectivity of n's neighbours in each



(a) Intersection of neighbourhoods



(b) Density of common neighbours



(c) Expanding Neighbourhoods

Figure 1. Determining community of node n: n should be assigned to community of 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, its common neighbours in L_1's intersection are more connected to each other, **c)** although n has the same number of common neighbours with L_1 and L_2 and both intersections are equally dense, it has more common neighbours with L_1 if we expand its neighbourhood boundary by one. In other words, its neighbours are more connected within the community of L_1.

community. One is more tempted to join a group where he or she has friends who already know each other (Figure 1(b)). And finally **c)** The depth of the neighbourhood. We further exploit the neighbourhood information by considering not only friends but also friends of friends, their friends and so on (Figure 1(c)).

1) Computation: We represent a network as a graph $G(V, E)$; V is set of nodes and E is set of edges. Let $e(u, v)$ denotes an edge between node u and v. Also consider $\aleph(v)$ denotes neighbourhood of node v, *i.e.* $u \in \aleph(v)$ *iff* there exists a path of length at most δ to that node from v. Here, δ is the neighbourhood threshold. If we set δ to 1, $\aleph(v)$ would be the immediate neighbours of node v or its direct friends. To compute the iCloseness we first score the neighbours of every node not only based on the depth of their neighbourhood (length of the path to them) but also based on how dense they are connected. Then we use scores of common neighbours between two nodes to calculate their iCloseness.

The **neighbours scoring**; every node u in $\aleph(v)$ is assigned with a neighbouring score relative to v, *i.e.* $ns(u, v)$. This score is propagated through unexplored edges as we expand the neighbourhood:

$$ns_1(u,v) = \begin{cases} 1 & \text{if } e(u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

$$E_1(v) = \{e(u,v) \in E\}$$

$$ns_l(u,v) = ns_{l-1}(u,v) + \sum_{e(u,m) \in E - E_{l-1}(v)} ns_{l-1}(m,v) \times \frac{e(u,m)}{\sum_i e(i,m)} \quad (1)$$

$$E_l(v) = E_{l-1}(v) \cup \{e(i,j) \in E \,|\, \exists\, e(j,k) \in E_{l-1}(v)\}$$

In this formula all the direct neighbours to v got an initial score of 1 and then the scores of non-direct neighbours are determined by spreading these initial scores through the unexplored edges. More precisely, the $ns_1(u, v)$ is initially 0 if u and v are not connected and otherwise it is 1 (or equal to the weight of the edge from u to v if weighted). And all the direct edges to v are marked as explored by adding to $E_1(v)$. Generally, $E_{l-1}(v)$ marks edges that have been explored in the previous levels and it is updated by adding the explored edges in each expansion level. More specifically, $E_1(v) = \{e(u, v) \in E\}$, direct edges to v, and at every level of expansion we expand $E(v)$ by one level, that is any edge $e(i, j)$ that is connected to one of the edges already in $E_{l-1}(v)$ is added to $E_l(v)$. See Figure 2 for an example of this relative scoring which encodes both depth of neighbouring/path length and connectivity of the neighbours/density.

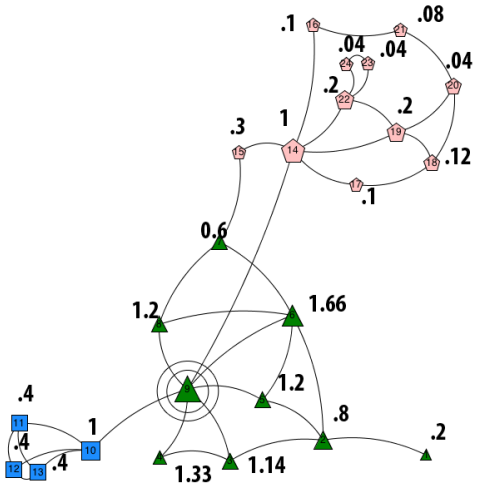


Figure 2. An example of scored neighbourhood relative to node 9. We can see that closer nodes to node 9 have higher neighbouring scores. Neighbours with more edges, those that are more densely connected to 9, also obtain higher scores (*e.g.* compare 6 and 8).

The ***iCloseness***, is defined based on neighbourhood intersections or common friends. The iCloseness of node v_1 and

v_2, is obtained as:

$$iCloseness(v_1, v_2) = \sum_{u \in \aleph(v_1) \cap \aleph(v_2)} ns(u, v_1) \times ns(u, v_2) \quad (2)$$

Figure 3 shows an example of the calculated iCloseness results which encodes the number of common neighbours and the connectivity of those neighbours and also the depth of the neighbourhood.

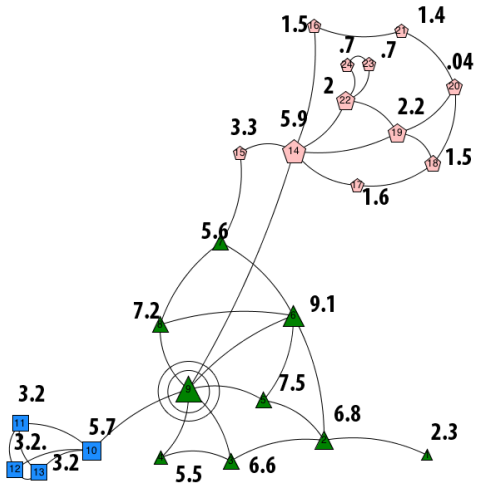


Figure 3. The same example as Figure 2 but marked with iCloseness of all nodes to node 9.

Note that *iCloseness* is symmetric for undirected networks but does not satisfy the triangle inequality. The neighbourhood threshold, δ, determines the maximum neighbourhood level which should be tuned based on the application. For example in social networks, the diameter of the whole network is very small even for the large scale networks, this is known as the six-degree sepration theoty or the small world phenomenon [22], [23]. Therefore a small number (*e.g.* 3) is sufficient for δ in such applications.

B. Top Leaders

Top Leaders first introduced in [5] is inspired by the well-known k-medoids clustering algorithm; however it works based on the relations between data points instead of their attributes. Similar in principle, this algorithm consists of choosing k representative nodes as leaders and then associating other nodes, the followers, to one of these leaders based on the relations/links between nodes to form communities. It iteratively 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. Here we adopted the original Top Leaders algorithm to use iCloseness when determining the community memberships of nodes. We call our version *iTop Leaders*. In this section we present in detail, this approach for detecting communities and elucidate the processes of selecting the initial leaders, associating followers to a leader, and electing new leaders.

1) Main Framework: iTop Leaders assumes that a community consists of a leader and the follower nodes associated with it; where the community leader is the most central member in its community. Some nodes in the network may not be associated with any of the leaders and thus do not belong to any community. These nodes are considered to be outliers if their centrality is low and hubs otherwise.

Algorithm 1 highlights the major steps of iTop Leaders algorithm. The first step is the selection of the initial k leaders (further described in Section III-B4). 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 or hubs (elaborated further in Algorithm 2), and second, when all nodes in the network are dealt with, a new leader is picked in each community.

Algorithm 1 iTop Leaders algorithm

Input: Social network G, integer k

```
initialize k leaders
repeat
  {finding communities}
  for all Node v ∈ G do
    if v ∉ leaders then
      find community of v {Algorithm 2}
    end if
  end for
  {updating leaders}
  for all ℓ ∈ leaders do
    ℓ ← arg max_{v ∈ Community(ℓ)} Centrality(v)
  end for
until there is no change in the leaders
```

2) Association of Nodes: Algorithm 2 depicts the process of associating a node with its *iClosest* leader which has been modified from the original version to use the *iCloseness*. As another modification, to find the *iClosest* leader to a given node, we initialize candidate leaders by considering only the leaders in its view, $\aleph(\aleph(v))$, that is neighbourhood of length $2 \times \delta$. These are the nodes that might *possibly* have common neighbours of depth δ with that node. This rational narrowing down of potential leaders also increases the speed of algorithm by many folds when applied to large networks and make the order of the algorithm independent to the number of communities or k. This gives us a major advantage over the original Top Leaders or the k-medoids algorithm.

We measure the *iCloseness* of the node v to each of its candidate leaders, to find the iClosest leaders. To detect

outliers in the network, we consider an outlier threshold (γ). Only leaders that are *iCloser* than this threshold to the node are considered. We also considered the case that the node is not *iClose* enough to any of the current leaders, but it has high centrality/power and should be considered as a hub not an outlier. Where originally, hubs were only considered as the nodes that follow more than one leader and sit on the intersection of communities.

Algorithm 2 Associate n to the *iClosest* leader

Input: Social network G, node v, set of k leaders

depth $\leftarrow$ 1
CanList $\leftarrow$ leaders $\cap$ $\aleph(\aleph(v))$
CanList $\leftarrow \underset{\substack{\ell \in \text{ CandList} \wedge \\ \text{iCloseness(v,}\ell) > \gamma}}{\arg\max}$ iCloseness(v, ℓ)
if |CanList| $= 0$ **then** {No candidate leader}
  if Centrality(v) $< \epsilon$ **then** {Noise}
    associate v as an outlier
  else {Powerful but free}
    associate v as a hub
  end if
else if |CanList| > 1 **then** {Many candidates}
  associate v as a hub
else {Only one candidate leader in CanList}
  associate v to the only leader in CanList
end if

3) Updating Leaders: The reassignment of leaders is simply the election of the node with the highest centrality in each community as its leader:

$$\ell \leftarrow \underset{v \in Community(\ell)}{\arg\max} \; Centrality(v)$$

This is because the centrality of nodes in a community measures the relative importance/popularity of a node within that group.

For a community C of size N, the degree centrality of a node v in C is:

$$Centrality(v) = \frac{degree(v, C)}{N - 1}$$

where $degree(v, C) = \sum_{u \in C} A_{vu}$ is the number of edges in C incident upon n. A represents adjacency matrix of the graph – A_{uv} is 1 if u is connected to v and 0 otherwise (if weighted, W_{vu} is used instead of A_{vu}).

4) Initialization: Community leaders are central nodes in their community. Therefore we choose the k most central nodes that none of them belong to the same community as initial leaders. To implement this strategy iTop Leaders start from the most central node, and add the next central one to the current set of leaders only if it is not too *iClose* to any of the current leaders (compared to a predefined threshold). A detailed comparison of different initialization methods for the Top Leaders is presented in [5].

IV. Experiments

We demonstrate, using two experiments, the application of our closeness measure. First in Section IV-A, we compare different closeness/distance measures when used with the same community mining algorithm – *i.e.* Top Leaders. Then in Section IV-B we compare iTop Leaders – in conjunction with iCloseness – against other well-known community mining methods.

The most common approach in evaluating community mining methods is to apply the algorithm to some well-known (typically small) real world data-sets for which the ground truth is known. The accuracy is then reported by comparing detected communities with the ground-truth using a measures of agreement. Typical measures are Adjusted Rand Index (ARI) [24] and Normalized Mutual Information (NMI) [25], ARI returns a value between -1 (*i.e.* no agreement at all) and 1 (*i.e.* full agreement) with expected value of 0 for agreement no better than random), where NMI is between 0(*i.e.* partitions are independent) and 1(*i.e.* partitions are identical). In our experiments we used five well-known real-world benchmarks: Karate and WKarate Club (weighted) by Zachary [26], Sawmill Strike data-set [27], NCAA Football Bowl Subdivision [2], and Politician Books from Amazon [28].

An alternative approach for evaluating a community mining method is testing it on synthesized networks that are generated with characteristics similar to that of real-world networks and with a built-in community structure.Here we used LFR benchmarks proposed by Lancichinetti *et al.* that have heterogeneous distribution over degrees and community sizes [29] and used in [25] to compare different community mining algorithms. In these benchmarks, node degrees and community sizes are derived from power law distributions while each node shares a fraction μ of its edges with the nodes of other communities. Here $0 \leq \mu \leq 1$, is called the mixing parameter [30].

We are also interested in further evaluation of the algorithm using large real networks. Unfortunately here, there is no explicit notion of ground-truth but we can check if the results are sound [14], [17]. We use the co-purchasing network of Amazon.com, collected in August 2003 [14]. It has $815,223$ nodes and $3,426,127$ edges where edges connect items that are frequently purchased together.

A. Comparing Measures

1) Comparing on real world benchmarks: Table I presents a comparison of our proposed closeness measure – iCloseness (iC) – against four other well-know distance/closeness measures for graphs: Shortest Path (SP), Adjacency Relation Distance (ARD), Neighbour Overlap Distance (NOD), and Pearson Correlation Distance (PCD) all surveyed in [30]. Here we used our community mining approach equipped with each of these measures to produce the results.

Table I
RESULTS OF TOP LEADERS ON REAL BENCHMARK DATA-SETS. IT COMPARES RESULTS OF TOP LEADERS WHEN USING DIFFERENT CLOSENESS/DISTANCE MEASURES.

dataset	measure	ARI	NMI
Karate (2 groups, 34 nodes, 78 edges)	Shortest Path	.669	.655
	Adjacency Relation	.771	.732
	Neighbour Overlap	.446	.383
	Pearson Correlation	.328	.324
	iCloseness	**1.**	**1.**
Strike (3 groups, 24 nodes, 38 edges)	Shortest Path	.935	.926
	Adjacency Relation	.903	.834
	Neighbour Overlap	.819	.763
	Pearson Correlation	.109	.307
	iCloseness	**1.**	**1.**
PolBooks (2 groups, 105 nodes, 441 edges)	Shortest Path	.647	.542
	Adjacency Relation	.630	.573
	Neighbour Overlap	.687	.585
	Pearson Correlation	.053	.157
	iCloseness	**.769**	**.696**
Football (11 groups, 180 nodes, 787 edges)	Shortest Path	.689	.559
	Adjacency Relation	.431	.753
	Neighbour Overlap	.970	.948
	Pearson Correlation	.082	.007
	iCloseness	**.996**	**0.989**

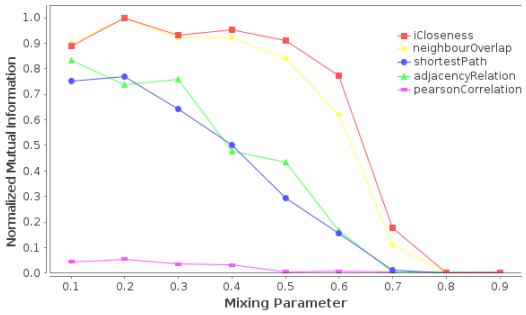


Figure 4. Comparison of different measures on LFR synthesized benchmarks. The horizontal axis represent datasets with different mixing parameter μ, and the vertical axis is the accuracy. Different curves stand for different measures.

We used the same parameters as well as initial leaders/centres for all methods. The only exception is the hub value for Politician Books data-set (which contains hubs) and the outlier threshold for the Football data-set (which contains outliers). These two parameters depend on the definition of the closeness measure and therefore, are adjusted based for each closeness measure. In all experiments we clearly obtained the best results using our *iCloseness* measure.

2) Comparing on synthesized benchmarks: Figure 4 compares different closeness/distance measures when applied to synthetic (larger scale) benchmarks. We generated LFR benchmarks with 1000 nodes, average degree of 20, maximum degree of 50 and different μ from .1 to .9, where communities have 20 to 100 nodes and the default values are used for exponents of the degree and the community size distributions (-2 and -1 respectively). These parameters are

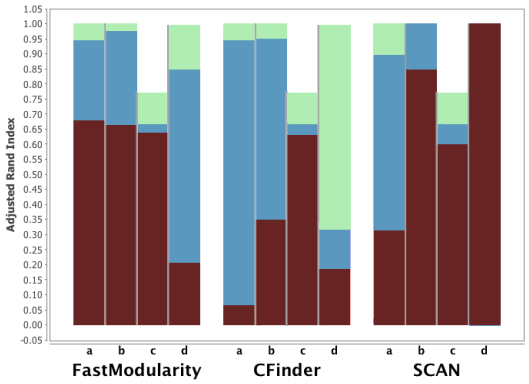


Figure 5. Comparison of iTop Leaders, iTL, with other approaches on real benchmark data sets – a) Karate, b) Strike, c) Politician Books and d) Football. For each data set/method, the red bar shows the accuracy of results obtained by that method, the blue bar shows the result of Top Leaders when seeded with the k obtained from the original method. The green bar is the ideal when Top Leaders is seeded with correct k.

the same as what Lancichinetti *et al.* used in [25] to compare different community mining algorithms (N=1000, B). We did not use the N=5000 due to computational constraints imposed by some of these measures *e.g.* SP is $O(n^3)$.

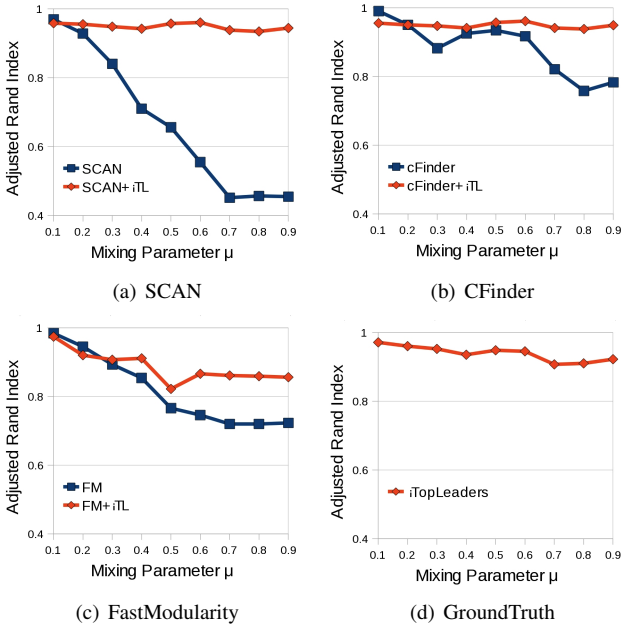
B. Comparing Algorithms

In this section, we compare iTop Leaders equipped with iCloseness, iTL, against three other well-known community mining approaches: SCAN [2], CFinder [31] and FastModularity [14].

1) Comparing on real world benchmarks: Figure 5 illustrates how iTop Leaders (iTL) improves the set of communities found by other methods. In this figure, red bars represent initial community mining algorithm, green bars represent iTop Leaders when seeded by the correct initial k. This comparison may not be fair as we fed iTop Leaders by prior knowledge of the network, *i.e.* number of communities (k). To have more reasonable comparison, we also compared all methods against iTop Leaders when fed by the k found by those methods (*i.e.* blue bars). We could see that iTop Leaders consistently improves on the other methods.

2) Comparing on synthesized benchmarks: To compare with other methods, we generated larger LFR networks with 5000 nodes and different μ from .1 to .9 (with the average degree of 15, the maximum degree of 50, community range of 200 to 500 nodes). Figure 6 highlights the robustness of iTop Leaders approach compared to FastModularity, CFinder and SCAN. The first three plots compare the ARI of each one of these three methods with iTop Leaders (seeded by the k from that method) on networks with different mixing parameter. In most cases and especially as μ increases, Top Leaders improves upon the other methods.

3) Comparing on large scale real-world data set: On the Amazon network, CFinder and SCAN did not terminate successfully, iTop Leaders terminated 10 times faster than



(a) SCAN (b) CFinder

(c) FastModularity (d) GroundTruth

Figure 6. The first three plots compare ARI of iTop Leaders (iTL) and other contenders as a function of mixing parameter μ. Figure 6(d) shows the results of iTop Leaders given the correct k.

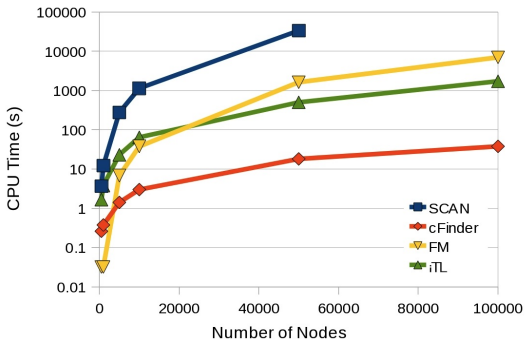


Figure 7. Comparison of running time of iTop Leaders and other methods for different number of nodes in the network.

FastModularity (for the same $k = 2303$ obtained by FastModularity). 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 communities compared to a randomized network. The modularity is zero when the portion of within community edges is not different from what we expect from a randomized network, and a value higher than 0.3 usually shows a significantly good partitioning [14]. In our experiment, modularity of iTop Leaders was 0.45 compared to .77 modularity of the result of FastModularity. Both rates guarantee that the communities hold strong modularity. However this does not show which algorithm is more accurate as the higher modularity does not ensure a higher accuracy (we have seen this in the previous experiments). Moreover our algorithm detected 89865 hubs which are not member of any specific community and this decreases the modularity significantly because modularity does not consider hubs.

4) Parameters: The only parameter of *iCloseness* is the neighbourhood threshold, δ. As described in III-A, δ determines the maximum neighbourhood level which should be set based on the application. As we discussed, for social networks a small diameter (*e.g.* 3) works fine.

Top Leaders algorithm on the other hand requires a few parameters – k, γ and ϵ. The main parameter is k, the number of desired communities. 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. Given this parameter, *Top leaders* always improves upon the contenders in terms of quality of the communities as demonstrated in our experiments. This is despite the fact that in most of the cases this k is not correct or even close to the correct value. FastModularity finds 12±6, CFinder finds 1182±464 and Scan finds 299±127 communities in the synthesized benchmarks when the average number of communities in the ground-truth is 33±5.

The outlier threshold, γ, and the hub threshold, ϵ, control the characteristics of the final output. If both are set to zero, the clustering output would detect no outliers and the clusters would be disjoint. By increasing the γ, more noise would be eliminated from the data and by increasing the ϵ, clusters' overlap increases. We set these two parameters to zero for all the experiments except for Politician Books and Football data sets, where finding hubs and outliers is desirable.

5) Complexity: The complexity of Top Leaders is $O(kn)$, the proof of which is similar to the one for k-medoids. Figure 7 illustrates a comparison for the running time of iTop Leaders, FastModularity, CFinder and SCAN. We have generated the LFR data sets for varying number of nodes ($\mu = .5$). All experiments are performed on the same machine and here we report the CPU time. The plot shows that iTop Leaders scales reasonably well and outperforms the speed of all contenders except CFinder.

V. Conclusions

We established a closeness measure, Intersection Closeness, to assess the proximity of a node to a community representative. This measure *iCloseness* is based on the theory of diffusion of innovation which states that the probability of joining a group depends on the number of existing friends in the group and their connectedness. We applied this measure to mine communities using the Top Leaders community mining algorithm, where this algorithm is adopted to use *iCloseness* in assigning nodes to communities. The experimental results of Top Leaders algorithm equipped with *iCloseness* show high accuracy and effectiveness achieved in both real and synthesized networks when compared against

commonly used closeness/distance measures as well as the state-of-the-art community mining methods.

REFERENCES

[1] S. Gregory, "An algorithm to find overlapping community structure in networks," in *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2007, pp. 91–102.

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

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999.

[4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *ACM SIGKDD*, 2006.

[5] R. Rabbany Khorasgani, J. Chen, and O. R. Zaiane, "Top leaders community detection approach in information networks," in *4th SNA-KDD Workshop on Social Network Mining and Analysis*, 2010.

[6] M. E. J. Newman, "Detecting community structure in networks," *Eur. Phys. J.B*, vol. 38, pp. 321–330, 2004.

[7] ——, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, 2006.

[8] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NIPS*, 2001, pp. 849–856.

[9] P. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral k-way ratio-cut partitioning and clustering," in *Proceedings of the 30th International Conference on Design Automation*, 1993, pp. 749–754.

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

[11] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, 2001, pp. 107–114.

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

[13] J. Chen, O. R. Zaïane, and R. Goebel, "Detecting communities in social networks using max-min modularity," in *SIAM Data Mining Conference*, 2009.

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

[15] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the USA*, 2002.

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

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

[18] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending modularity definition for directed graphs with overlapping communities," 2008. [Online]. Available: http://arxiv.org/abs/0801.1647

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

[20] D. Strang and S. Sarah A, "Diffusion in organizations and social movements: From hybrid corn to poison pills," *Annual Review of Sociology*, vol. 24, no. 1, pp. 265–290, 1998.

[21] P. Sarkar and A. Moore, "Dynamic social network analysis using latent space models," *ACM SIGKDD Explorations: Special Edition on Link Mining*, vol. 7, no. 2, pp. 31–40, 2005.

[22] L. Getoor and C. P. Diehl, "Link mining: a survey," *ACM SIGKDD Explorations*, 2005.

[23] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, no. 1, pp. 60–67, 1967.

[24] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *ICANN (2)*, 2009, pp. 175–184.

[25] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E*, vol. 80, no. 5, p. 056117, Nov 2009.

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

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

[28] V. Krebs, "Books about us politics," http://www.orgnet.com/.

[29] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, p. 046110, 2008.

[30] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, 2010.

[31] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, pp. 814–818, 2005.