2011 11th IEEE International Conference on Data Mining

Entropy-Based Graph Clustering: Application to Biological and Social Networks

Edward Casey Kenley
Department of Computer Science
Baylor University
Waco, TX, USA
casey_kenley@baylor.edu

Young-Rae Cho
Department of Computer Science
Baylor University
Waco, TX, USA
young-rae_cho@baylor.edu

***Abstract*—Complex systems have been widely studied to characterize their structural behaviors from a topological perspective. High modularity is one of the recurrent features of real-world complex systems. Various graph clustering algorithms have been applied to identifying communities in social networks or modules in biological networks. However, their applicability to real-world systems has been limited because of the massive scale and complex connectivity of the networks. In this study, we exploit a novel information-theoretic model for graph clustering. The entropy-based clustering approach finds locally optimal clusters by growing a random seed in a manner that minimizes graph entropy. We design and analyze modifications that further improve its performance. Assigning priority in seed-selection and seed-growth is well applicable to the scale-free networks characterized by the hub-oriented structure. Computing seed-growth in parallel streams also decomposes an extremely large network efficiently. The experimental results with real biological and social networks show that the entropy-based approach has better performance than competing methods in terms of accuracy and efficiency.**

***Keywords*-graph mining; graph clustering; complex systems; biological networks; social networks; graph entropy**

I. Introduction

A complex system is defined as a dynamically evolving network that has a large number of objects with complicated connectivity among them. Typical real-world examples include social networks, biological networks, telecommunication networks, and the World Wide Web. These systems are often described in a graph representation. Social networks represent individuals (or organizations) as nodes and social relationships between them as edges. Biological networks describe a series of biochemical reactions or biophysical interactions between molecular components.

These complex systems have been widely studied, with efforts focusing on characterizing their structural behaviors from a topological perspective. The scale-free network model is one such example with intriguing features, e.g., a power-law degree distribution having a heavy tail [1]. This observation leads to the interpretation that these graphs have a large number of low-degree peripheral nodes which are linked to a few high-degree hubs. Another common property of complex systems is high modularity. In general, they have higher clustering coefficients than a random graph. This means the network can decompose into a set of sub-graphs based on connectivity.

Measuring modularity and developing scalable clustering algorithms are key issues in current study of complex systems. A cluster as a sub-graph, also called a community in social networks and a module in biological networks, is a group of objects that closely communicate with each other. For example, in a global-scale Facebook network, a cluster becomes a potential group having the same interests or the same backgrounds. In a genome-wide protein-protein interaction network, a cluster is expected to be a group of proteins performing the same biological function. In recent years, various clustering algorithms with graph-theoretic modeling have been applied to identifying communities or modules in complex systems. However, the applicability of the previous graph clustering approaches to real-world networks has been limited because of the following challenges:

- Real-world complex systems are mostly sized on a massive scale. Mining the networks requires scalability.
- Complexity in the network structure imposes a limitation on accurate and efficient mining.

A novel strategy to effectively extract valuable information hidden in large-scale complex networks is urgently needed. In this study, we exploit a new information-theoretic model for graph clustering. We have applied the concept of entropy, or structural instability, to graph representations. We assume that a network has the lowest entropy, or the highest structural stability, when it is composed of sub-graphs that are isolated from each other. Random reconnection makes a transition from this stable modular state to a disorganized state and increases entropy of the network. In contrast, loss of entropy represents an increase in modularity.

The entropy-based graph clustering algorithm first selects a random seed as the initial node and forms a seed cluster by including all neighbors of the seed. Next, the seed cluster shrinks and grows to minimize the graph entropy by iteratively removing and adding the nodes on the border of the cluster. This seed growth stops when a locally optimal boundary is found. The process of selecting a seed and generating an optimal cluster is repeatedly performed, and the algorithm finally produces a set of clusters.

1550-4786/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDM.2011.64

We additionally design and analyze modifications of the entropy-based algorithm for performance improvement. We assign a priority to selecting a seed and also to selecting a node on the cluster border during seed growth. The process to select cores as seeds is well applicable to the scale-free networks characterized by the hub-oriented structure. We also include a post-processing step to filter out less accurate clusters that have higher graph entropy than a given threshold. Furthermore, we compute seed growth in parallel streams for decomposing an extremely large network efficiently. We test these improvements on real-world social networks and biological networks. We demonstrate that this entropy-based approach has better performance than other competing methods in terms of accuracy and efficiency.

II. Entropy-Based Clustering

A. Graph Entropy

Graph Entropy is a new information-theoretic definition to assess modularity of a graph. Suppose an undirected, unweighted graph $G(V, E)$ is decomposed into a set of clusters. A cluster is considered an induced subgraph $G'(V', E')$ of G, which has dense intra-connections within G' and sparse interconnections between G' and $(G - G')$.

Given a cluster $G'(V', E')$, we define the inner links of a vertex v as the edges from v to the vertices in V'. The outer links of v are defined as the edges from v to the vertices not in V'. $p_i(v)$ denotes the probability of v having inner links.

$$p_i(v) = \frac{n}{|N(v)|},$$

where $|N(v)|$ is the total number of neighboring vertices of v, and n is the number of the neighboring vertices of v that are in V'. Similarly, $p_o(v)$ denotes the probability of v having outer links.

$$p_o(v) = 1 - p_i(v).$$

Definition 1. **Vertex Entropy**. Given a cluster, the entropy $e(v)$ of a vertex v is defined based on the probability distribution of its inner links and outer links.

$$e(v) = -p_i(v) \log_2 p_i(v) - p_o(v) \log_2 p_o(v).$$

Definition 2. **Graph Entropy**. Given a cluster, the entropy $e(G)$ of a graph $G(V, E)$ is then defined as the sum of the entropy of all vertices in G.

$$e(G) = \sum_{v \in V} e(v).$$

Figure 1 (a) illustrates an example of a graph $G(V, E)$ containing a cluster $G'(V', E')$ where $V' = \{a, b, c, d\}$. For the vertex a, all the neighbors are in the cluster G', thus $p_i(a) = 1$, $p_o(a) = 0$, and its entropy $e(a) = 0$ by Definition 1. For the vertex g, all the neighbors are outside of G', thus $p_i(g) = 0$, $p_o(g) = 1$, and $e(g) = 0$. However, for the vertex f, since one neighbor is in G' and the other is not, $p_i(f) = p_o(f) = 0.5$ and $e(f) = 1$. This definition of entropy indicates that a vertex has the highest entropy if its directly connected neighbors are evenly divided into inside and outside of the cluster. Also, only the vertices that have both inner links and outer links have entropy greater than 0.

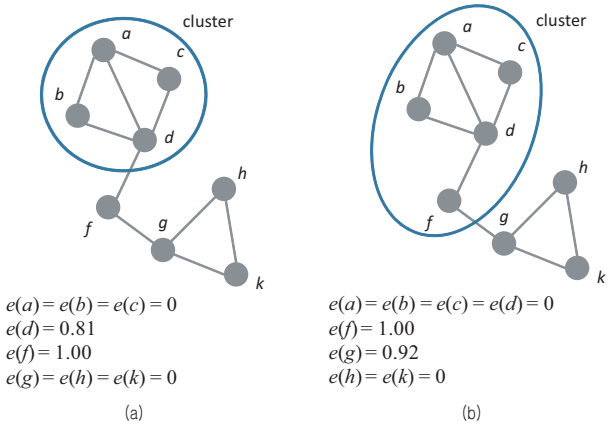

Figure 1. Example of vertex entropy and graph entropy measurement. In (a), given a cluster with a, b, c and d, entropy of the graph is 1.81. However, in (b), if the vertex f is added into the cluster, graph entropy increases to 1.92 because the cluster quality decreases.

The quality of a cluster can be evaluated by connectivity, i.e., higher quality as denser intra-connections and sparser interconnections. The graph entropy definition is formulated to measure the cluster quality effectively. A graph with lower entropy indicates the vertices in the cluster have more inner links and less outer links. In Figure 1 (a), the graph entropy $e(G)$ is 1.81 by Definition 2. However, in Figure 1 (b), if the vertex f is added into the cluster, the graph entropy increases to 1.92, thus the cluster quality decreases.

B. Entropy-Based Clustering Algorithm

The entropy-based graph clustering algorithm repeatedly finds a locally optimal cluster with minimal graph entropy. A high-level description of the algorithm is given below.

1) Select a random seed vertex. Form a seed cluster including the selected seed and its neighbors.
2) Iteratively remove any of the seed neighbors to minimize graph entropy.
3) Iteratively add vertices on the outer boundary of the cluster to minimize graph entropy.
4) Output the cluster. Repeat steps 1, 2 and 3 until all vertices have been clustered.

In step 1, a set of seed candidates is managed. When a cluster is generated in step 4, the vertices in the cluster are removed from the candidate set, i.e., they are excluded from being selected as a seed vertex for any subsequent cluster. This is the way to avoid generating duplicate clusters.

In step 2, each seed neighbor is greedily checked if its removal decreases graph entropy. Similarly, in step 3, each

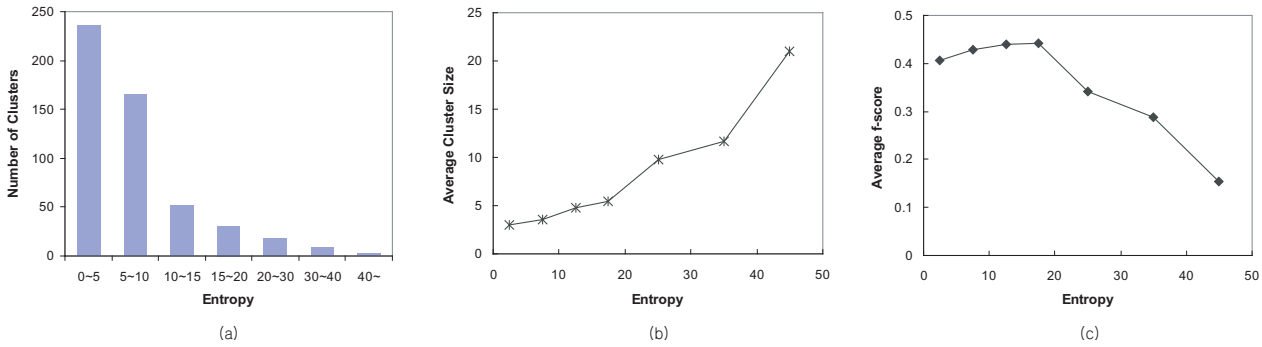

Figure 2. (a) The distribution of output clusters with respect to graph entropy. (b) The relationship between the graph entropy and size of the clusters. (c) The relationship between the graph entropy and f-scores of the clusters.

vertex on the outer boundary is added into the cluster if graph entropy decreases. The vertices on the outer boundary mean the vertices outside the cluster but having inner links to the cluster members. The steps 1, 2 and 3 detect an optimal cluster with the lowest graph entropy.

These three steps are repeated to yield a set of clusters. During the repetition, the vertices in the previously generated clusters should remain in the graph. This means the vertices in a prior cluster can be members of the subsequent clusters. This algorithm is thus able to generate overlapping clusters.

III. Application to Biological Networks

A. Test Configuration

We tested the entropy-based clustering in a real biological network. We downloaded the genome-wide protein-protein interaction data of yeast from the Database of Interacting Proteins (DIP) [2], which have 4928 proteins as vertices and 17186 interactions as edges. Proteins interact with each other to form a protein complex. We thus evaluated clustering accuracy by comparing the output clusters to known protein complexes. The yeast protein complex data, as ground truth, were obtained from MIPS [3].

To measure clustering accuracy, we used the f-score, the harmonic mean of recall and precision.

$$Recall = \frac{|X \cap P_i|}{|P_i|}$$

and

$$Precision = \frac{|X \cap P_i|}{|X|},$$

where X is a set of vertices in an output cluster and P_i is a set of proteins in a protein complex. For each output cluster, we searched the best match from the protein complexes in regard to f-scores. We then averaged the f-scores of the best matches across all output clusters.

Implementing the entropy-based algorithm with the protein interaction network, we obtained 501 clusters. The average size of the clusters is approximately 4. We evaluated the effect of graph entropy on clustering accuracy. Figure 2 (a) shows the distribution of the clusters with respect to graph entropy. Only 5% out of 501 clusters have graph entropy greater than 20. We also investigated the relationship between the cluster size and graph entropy in Figure 2 (b). These figures clearly show that a few clusters with high graph entropy are larger than the others. Finally, Figure 2 (c) shows clustering accuracy with respect to graph entropy. The average f-score of all output clusters is 0.414. The clusters with graph entropy lower than 20 have acceptable accuracy on average. However, when the graph entropy is greater than 20, the cluster accuracy rapidly decreases as entropy increases. Because of the observation that a few clusters with relatively high entropy result in low accuracy, we removed such clusters above a specific entropy threshold as a post-process. When we use 20 as the entropy threshold, the average f-score increased to 0.420 as shown in Table I.

B. Improvement of Randomness

The main process in the entropy-based algorithm relies on randomness. Each seed vertex is selected in a random order. During seed growth, each neighboring vertex on the cluster boundary is also selected in a random order. We thus revised the random selection to deterministic processes. First, in seed selection, a core (or hub) of the potential cluster is an optimal candidate as a seed. There are several measures to find the cores (or hubs) in a graph. For example, vertex degree in real biological networks has a strong positive correlation with functional essentiality. The clustering coefficient [4] of a vertex, which represents the proportion of connections among neighbors of the vertex, is also a common measure to find cores of densely connected sub-graphs. We assigned a higher priority in seed selection to the vertex of higher degree or higher clustering coefficient. Next, during seed growth, a neighboring vertex of a cluster can be selected in a specific order. Intuitively, the one with the lower vertex entropy should be added earlier into the cluster. As a deterministic greedy approach, selecting the neighbor with the lowest entropy explicitly gives a higher chance to achieve the local optimum.

Table I
CLUSTERING RESULTS AND AVERAGE f-SCORES OF THE ENTROPY-BASED APPROACH AFTER APPLYING IMPROVEMENTS TO REDUCE RANDOMNESS. AS A POST-PROCESS, THE CLUSTERS WITH GRAPH ENTROPY GREATER THAN A THRESHOLD OF 20 WERE REMOVED.

improvements	before post-process			after post-process		
	# clusters	avg size	avg f-score	# clusters	avg size	avg f-score
original algorithm	501	3.82	0.414	474	3.50	0.420
degree-based seed selection	442	4.02	0.416	413	3.60	0.423
coef.-based seed selection	551	3.73	0.415	522	3.39	0.422
degree-based seed selection AND lower entropy-based seed growth	442	4.03	0.416	413	3.61	0.424
coef.-based seed selection AND lower entropy-based seed growth	551	3.74	0.415	522	3.39	0.422

Table I shows increase of clustering accuracy when a seed is iteratively selected in the decreasing order of degree or clustering coefficients. Comparing to the results of the original algorithm, the degree-based seed selection generated a smaller number of larger clusters. Selecting each seed in the order of clustering coefficients, we obtained a larger number of smaller clusters. The degree-based seed selection shows a better improvement in average f-scores than the clustering coefficient-based seed selection. Moreover, when we applied the post-processing step of filtering out the clusters with the graph entropy greater than a threshold of 20, the average f-score increased up to 0.423. Assigning a priority on seed growth by the lowest vertex entropy makes the algorithm deterministic. However, as shown in Table I, it did not improve clustering accuracy.

C. Overlap Analysis

Two or more clusters overlap if they have common members. The real-world complex systems typically include a significant number of overlapping clusters. In social networks, an individual might have membership into two or more different communities. In biological networks, a molecular component might perform multiple functions.

Previous partition-based or hierarchical clustering methods are not able to generate overlapping clusters. However, the entropy-based approach can find the clusters that share a vertex v if v has the same number of links to each of them. In the experiment with the yeast protein interaction network, 221 overlapping cluster pairs were produced. Around 10% of the vertices were involved in the overlaps. The average overlap size, i.e., the number of vertices on each overlapping event, was 2.23. The post-process decreased the number of overlapping cluster pairs by around 30% although it removed only 5% of the output clusters. The number of vertices in the overlaps also decreased. This indicates that the clusters deleted by the post-process include a larger ratio of overlaps. In other words, a significant proportion of the overlaps occur when clustering is inaccurate.

D. Accuracy Analysis

We compared the performance of the entropy-based clustering algorithm with other competing methods: MCL [5] and CNM [6]. Markov Clustering (MCL) algorithm simulates random walks within a Markov matrix as an input graph by repeatedly alternating two operators: expansion and inflation. This process continues until there is no further change in the matrix and finally finds the best partition of the graph. The CNM algorithm was proposed to efficiently detect communities in very large social networks. It performs a bottom-up greedy optimization to maximize the modularity. The modularity Q of a cluster i was introduced as

$$Q = \sum_i (e_{ii} - a_i^2),$$

where e_{ii} is the fraction of the number of links between two vertices within the cluster i, and a_i^2 is the fraction of the number of all links starting from any vertices in i.

Table II
CLUSTERING RESULTS AND ACCURACY OF THE ENTROPY-BASED APPROACH, MCL, AND CNM IN A BIOLOGICAL NETWORK.

method	# clusters	avg size	avg f-score
Entropy-based	413	3.61	0.424
MCL	428	5.42	0.419
CNM	40	58.55	0.391

Table II shows the comparison of clustering results and accuracy of the three algorithms with the yeast protein interaction network. MCL and the entropy-based method produced a sufficient number of clusters with the appropriate size for real protein complexes. However, CNM generated a smaller number of substantially larger clusters. When the best condition in Table I is chosen, the entropy-based approach has higher accuracy than MCL and CNM.

IV. APPLICATION TO SOCIAL NETWORKS

A. Test Configuration

To test the entropy-based clustering in social and telecommunication networks, we downloaded three datasets: AS link network [7], YouTube video link network [8], and MySpace social network [9]. An autonomous system (AS) is the fundamental unit of routing policy on the internet. All networks that belong to the same AS are administered by the same entity, e.g., a research university or enterprise business. Decisions about where to route network traffic are

Table III
NETWORK SIZE AND DENSITY OF AS LINK NETWORK, YOUTUBE VIDEO LINK NETWORK AND MYSPACE SOCIAL NETWORK.

dataset	# vertices	# edges	density (%)
AS links	45,744	323,009	0.031
YouTube	321,683	505,845	0.001
MySpace	100,000	6,854,231	0.137

based upon the source AS and destination AS. The AS link network is composed of autonomous systems represented by nodes and physical links between them represented by edges. The YouTube video link network was formed by the video files uploaded to YouTube as nodes. Each video file has links to a list of related videos as edges. The MySpace dataset was obtained by crawling the MySpace website, one of the social networking services. This network was formed by making each account a node and forming an edge wherever a friend relationship exists. Table III shows the size and density of the three networks. The AS link network is smaller than the others, and the YouTube network is the largest but has low density. The MySpace network is mid-sized and very dense.

We implemented the entropy-based clustering algorithm on these large-scale networks with different features to find a set of clusters as potential communities. All social networks were pre-processed, i.e., all self-loops and multi-edges were removed from the datasets, and only the largest connected component was considered. In other words, the vertices that were unreachable from the largest connected sub-graph were removed from each dataset. The accessory components that were pruned from these graphs were quite small.

B. Parallelization

The algorithms to handle a vast amount of data should be considered in light of its amenability to parallel implementation. The clustering algorithms based on seed growth naturally lend themselves to parallelization. Seeds can be grown independently from one another, allowing this computation step to occur simultaneously on several processors. Under ideal conditions, each processor would reduce the computation time by the same margin. However, the serial algorithm of the entropy-based clustering, described in Section II-B, does not have to contend with multiple seed growth instances concurrently producing different clusters that share at least one common vertex.

To avoid wasteful computations in the parallel version of the entropy-based approach, we designate parts of steps 1 and 4 of this algorithm as the critical section. In step 1, each seed growth instance must atomically select the seed vertex that it will grow. Failure to do so will result in the creation of an exact duplicate cluster, a complete waste of resources. Step 4 also accesses a shared resource when removing the members of a cluster from the set of candidate seeds. Guaranteeing mutual exclusion to this data structure also reduces the chance that a redundant cluster will be grown.

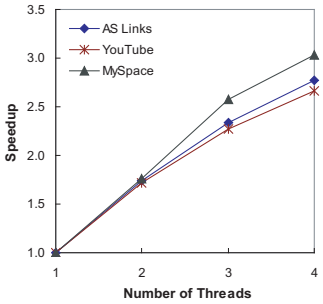

Figure 3. Speedup of parallelizing the entropy-based approach. The speedup by x threads represents the ratio of the elapsed time with 1 thread to the elapsed time with x threads.

We implemented the multithreaded version of the entropy-based clustering algorithm in Java. As the number of threads increased, runtime explicitly decreased with similar rates on all three networks. The speedup by parallelization is described in Figure 3. The speedup by x threads represents the ratio of the elapsed time with 1 thread to the elapsed time with x threads. The result indicates that the entropy-based algorithm with 4 threads ran 3 times faster than the serial version in the MySpace network. We can achieve higher speedup in the implementation with a denser network.

C. Efficiency Analysis

Table IV shows the trends in the clustering results produced by the entropy-based algorithm, MCL and CNM. In general, our entropy-based approach generated a large number of small-sized clusters. On the other end of the performance spectrum, we note that CNM tends to produce larger, but fewer clusters. MCL results tend to vary based on the dataset that is being examined. On the AS link and YouTube graphs, clustering results are closer to our algorithm. However, in the MySpace dataset, MCL produces even larger clusters than CNM, eliminating any conformity with a simple trend.

The runtime of the three algorithms is described in Table V. For the entropy-based clustering, we show the results with 1 thread and 4 threads. Comparing the runtime, we see that our approach produces the quickest results for the AS link and YouTube datasets (even with just a single thread). In contrast, the MySpace dataset was decomposed the fastest by MCL. The margin of performance on MySpace between our solution and the competing solutions likely lies in the density of the graph. We note that the MySpace graph is much denser than the other two graphs. Our algorithm iterates over adjacent edges as it seeks to expand the cluster boundary. Heavier nodes in a cluster will create more candidates for expansion than lighter nodes, and these outer boundary candidates add significant time to the graph entropy calculation. We can thus conclude that our method is

Table IV
CLUSTERING RESULTS OF THE ENTROPY-BASED APPROACH, MCL, AND CNM IN THREE REAL-WORLD NETWORKS.

method	AS link		YouTube		MySpace	
	# clusters	avg cluster size	# clusters	avg cluster size	# clusters	avg cluster size
Entropy-based	824	63.1	25,897	8.8	6,505	6.2
MCL	979	46.7	23,001	14.0	117	853.8
CNM	79	578.5	886	363.1	195	512.8

Table V
RUNTIME (IN SECONDS) OF THE ENTROPY-BASED APPROACH, MCL, AND CNM IN THREE REAL-WORLD NETWORKS.

method	AS link	YouTube	MySpace
Entropy (1 thread)	162	75	50,028
Entropy (4 threads)	58	28	16,519
MCL	6,983	317	3,764
CNM	414	118	5,334

highly suitable for clustering sparse graphs, even very large ones, but dense graphs slow the clustering process down.

D. Accuracy Analysis

Because we were not able to obtain the real community information as ground truth, we used a statistical metric for estimating the clustering accuracy. Under the null hypothesis that the set of vertices V' in a cluster $G'(V', E')$ is chosen randomly from the original graph $G(V, E)$, the probability of observing that a vertex v is directly connected to a vertex in V' is described as the p-value in a cumulative hypergeometric distribution as follows:

$$p(v) = \sum_{i=|N(v)\cap V'|}^{\min(N(v),|V'|)} \frac{\begin{pmatrix} |N(v)| \\ i \end{pmatrix} \times \begin{pmatrix} |V| - |N(v)| \\ |V'| - i \end{pmatrix}}{\begin{pmatrix} |V| \\ |V'| \end{pmatrix}},$$

where $N(v)$ denotes a set of direct neighboring vertices of v, and $|V'|$ is the size of the vertex set V'. A low p-value in this formula indicates that the neighbors of v are mostly included in the cluster G'. The quality of the connection pattern of v is then defined as the negative log of p-value, $-\log p(v)$, which is called the p-score. This measure evaluates the statistical significance of each member of the cluster in terms of connectivity.

We computed the quality of a cluster $G'(V', E')$ by averaging the p-scores of all vertices in V', and then measured the accuracy of a clustering algorithm by averaging the p-scores of all output clusters. On the AS link dataset, the entropy-based clustering algorithm had an average p-score of 5.49, whereas MCL's average p-score was 4.31. This potentially demonstrates that our approach has higher accuracy than competing algorithms even in real-world large-scale networks.

V. CONCLUSIONS

We proposed a novel information-theoretic approach for graph clustering. This method is designed by new definitions, vertex entropy and graph entropy, derived from connectivity patterns. It adopts a greedy-style algorithm to find each locally optimal cluster. This approach tackles challenges of previous graph clustering methods on large-scale complex networks.

The entropy-based approach has effectively competed with commonly-used graph clustering algorithms. It performed with high accuracy and efficiency in uncovering functional modules from complex biological networks and communities from extremely large social networks. The proposed approach thus has many potential applications in diverse areas such as computational systems biology, social network analysis, national security, and internet flow analysis.

REFERENCES

[1] Barabasi, A.-L. and Oltvai, Z.N., "Network biology: understanding the cell's functional organization," *Nature Reviews: Genetics*, vol. 5, pp. 101–113, 2004.

[2] Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U. and Eisenberg, D., "The database of interacting proteins: 2004 update," *Nucleic Acids Research*, vol. 32, pp. D449–D451, 2004.

[3] Mewes, H.W., et al., "MIPS: analysis and annotation of genome information in 2007," *Nucleic Acids Research*, vol. 36, pp. D196–D201, 2008.

[4] Watts, D.J. and Strogatz, S.H., "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[5] Van Dongen, S., "A new clustering algorithm for graphs," National Research Institute for Mathematics and Computer Science in the Netherlands, Tech. Rep. INS-R0010, 2000.

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

[7] Oliveira, R., Pei, D., Willinger, W., Zhang, B. and Zhang, L., "Quantifying the completeness of the observed internet AS-level structure," UCLA, Tech. Rep. TR-080026-2008, 2008.

[8] Cheng, X., Dale, C. and Liu, J., "Statistics and social network of YouTube videos," in *Proceedings of 16th IEEE International Workshop on Quality of Service (IWQoS)*, 2008, pp. 229–238.

[9] Ahn, Y.-Y., Han, S., Kwak, H., Moon, S. and Jeong, H., "Analysis of topological characteristics of huge online social networking services," in *Proceedings of 16th International Conference on World Wide Web (WWW)*, 2007, pp. 835–844.