Mining Parameters that Characterize the Communities in Web-like Networks

Narsingh Deo, *Fellow, IEEE*, and Aurel Cami

***Abstract*—Community mining in large, complex, real-life networks such as the World Wide Web has emerged as a key data mining problem with important applications. In recent years, several graph theoretic definitions of community, generally motivated by empirical observations and intuitive arguments, have been put forward. However, a formal evaluation of the appropriateness of such definitions has been lacking. We present a new framework developed to address this issue, and then discuss a particular implementation of this framework. Finally, we present a set of experiments aimed at evaluating the effectiveness of two specific graph theoretic structures—alliance and near-clique—in capturing the essential properties of communities.**

***Index Terms*—graph mining, statistical data mining, random graphs, cyber communities**

I. Introduction

Interest on the structure and function of large, complex, real-life networks such as the World Wide Web (Web) and the Internet has surged dramatically in recent years. Several experimental studies have shown that such networks display a number of properties that markedly distinguish them from the classical Erdös-Rényi random graphs. These properties include a heavy-tailed degree distribution, a high degree of clustering, and a globally-sparse, locally-dense topology—referred to as the *community structure* [20]. We use the term *web-like* to refer to the networks that display these properties. Our focus in this paper is on the community structure of these networks.

Due to a wide array of potential applications, ranging from search engines and topic directories to graph compression and network security, community mining in web-like networks has attracted a great deal of attention. A number of graph theoretic definitions of community, generally motivated by empirical observations and intuitive arguments, have appeared in literature [7], [14]. However, a formal evaluation of the appropriateness of such definitions has been lacking. To fill this void, we have developed a framework for evaluating the suitability of a particular definition of community. This framework consists in estimating through sampling techniques the concentration in web-like networks of a parameterized definition of community and then deducing the statistical significance of this concentration in various ranges of the involved parameters by contrasting with an appropriately defined random graph.

Community-mining in web-like networks may be viewed as a specific line of investigation within the wider research area of *graph mining*. The main thrust of the work in this area has been the discovery of computationally efficient algorithms for mining special types of subgraphs such as *complete bipartite subgraphs* [14], [9], *quasi-cliques* [1], *connection subgraphs* [6], and *trees* [23]. An interesting recent trend is the simultaneous mining of multiple graph datasets; this setting of the graph-mining problem, which has been called *graph-transaction mining* [15] or *cross-graph mining* [20], might boost the detection of certain patterns which are difficult to detect through the analysis of data from a single graph.

Web-like networks tend to be massive, with sizes varying from several thousands of nodes and edges (e.g., biological networks) to millions and even billions of nodes and edges (e.g., the Web). To deal with such large datasets, the focus in pattern-mining is moving from algorithms that assume an internal representation of data in the main memory [12] to algorithms that assume a semi-external [1] or *external, disk-resident* data representation [6], [9], [21].

Another new and appealing direction is the analysis of *dynamic* graph datasets made up of sequences of snapshot acquired from a network over an extended period of time. The recent results presented in [16] show that such analysis might enable us to achieve a finer understanding of network topology which in some cases may even contradict the current beliefs about the evolution of such networks (e.g., in [16] it was found that, contrary to the widespread belief, the diameter of several web-like networks shrinks with time rather than grow). An interesting related problem is that of *link prediction* [17]—the accurate prediction of the future network evolution based on past network data.

Although the vast majority of the research in the area of graph mining deals with the problem of mining specific patterns in graphs, work on a complementary aspect, namely the *mining of graph properties* has also appeared [10]. The topic of [10] is similar in scope to that of the present paper, in that both works focus on mining graph properties rather than subgraphs which satisfy certain constrains.

Next, we continue with a discussion of some graph theoretic concepts which are used in the remainder of the

Manuscript received December 30, 2005. Both authors are with the School of Computer Science, University of Central Florida, Orlando, FL 32816 (First author's phone: (407) 823-6336, e-mail: deo@cs.ucf.edu; second author's e-mail: acami@cs.ucf.edu).

paper.

II. PRELIMINARIES

A. Notation

Table I summarizes the notation used in this paper.

TABLE I
LIST OF SYMBOLS

Symbol	Meaning
A_α	An α-alliance
C_{rand}	The concentration of a subgraph in the graph G_{rand}
C_{real}	The concentration of a subgraph in a real web-like network
$\mathbb{E}$	Expectation
$G(V,E)$	A graph with set of nodes V and set of edges E
$G[S]$	The subgraph induced by the subset of nodes S
$G_{n,p}$	The classical Erdös-Rényi random graph
G_{rand}	A random graph used as the baseline for determining the statistical significance of subgraph concetration
K_n	The complete graph
$K_{m,n}$	The complete bipartite graph
n	The number of nodes of a graph
m	The number of edges of a graph
N_α	An α-near-clique
$N(u)$	The open neighborhood of node u —the set of nodes adjacent u
$N[u]$	The closed neighborhood of node u : $N[u]=N(u)\cup\{u\}$
$T(V,E)$	A tree with set of nodes V and set of edges E

The *clustering coefficient* of a node u is defined as the probability that two random neighbors of u are neighbors themselves [22]. The term *size* in this paper refers to the number of nodes of a graph (sometimes the number of nodes is referred to as the *order* of the graph while the term *size* is reserved for the number of edges). Unless otherwise noted, all graphs in this paper are assumed to be *simple*.

B. Subgraph concentration

Definition 1. The *concentration* of a subgraph of type t and size s in a given graph is defined as the fraction of connected subgraphs induced by node-subsets of cardinality s, that are of type t.

C. Alliances and Near-Cliques

A "web community" was defined in [7] (p. 1) as "*…a set of sites that have more links (in either direction) to members of the community than to non-members*". The same definition of community has been subsequently adopted by several other authors. This intuitive definition is rooted in the study of social networks and as it turns out, a new name—*defensive alliance*—has been given in graph theory to sets of nodes satisfying the just mentioned property [13]. Formally, a subset of nodes S in a graph G is called a defensive alliance if $|N[v]\cap S|\geq|N[v]-S|$ for all nodes $v\in S$. Some other types of alliance such as *offensive* and *powerful* have also been defined [3], [13].

The following definition introduces a generalization of defensive alliance:

Definition 2. Let $G=(V,E)$ be a graph and $\alpha\in[0,1]$ a real number. An $\alpha\text{-}alliance$ is defined as a subset of nodes A_α such that every node u in A_α satisfies the inequality $|N[u]\cap A_\alpha|\geq\alpha|N[u]-A_\alpha|$.

Next, we introduce a new definition of a community called *near-clique*. Our motivation stems from the following two considerations: (i) Several alliance-mining problems have been shown to be NP-hard [3], [8]. Therefore it becomes necessary to investigate the existence of alternative definitions of community which render community-mining amenable to polynomial-time algorithms; (ii) Clustering coefficient seems to capture well the intuitive notion of communities as groups of nodes that share a common theme [4], [5].

Definition 3. Let $G=(V,E)$ be a graph and $\alpha\in[0,1]$ a real number. An $\alpha\text{-}near\text{-}clique$ is defined as a subset of nodes N_α such that the clustering coefficient with respect to (w.r.t.) the induced subgraph $G[N_\alpha]$ of each node u in N_α is greater than or equal to α.

As an illustration, we list some well-known graphs and the ranges of α where each of these graphs forms an $\alpha\text{-}near\text{-}clique$: (a) The complete graph K_n forms an $\alpha\text{-}near\text{-}clique$ for all $\alpha\leq 1$; (b) When $k=2$, the *k*-nearest neighbor lattice [22], forms an $\alpha\text{-}near\text{-}clique$ for all $\alpha\leq 0.5$; (c) The complete bipartite graph $K_{m,n}$ does not form an $\alpha\text{-}near\text{-}clique$ for any $\alpha>0$.

A strikingly similar term—*quasi-clique*—has appeared in the recent literature with two different meanings: In [1] a $\gamma\text{-}quasi\text{-}clique$ was defined as a subset of nodes S such that the number of edges in the induced subgraph $G[S]$ is at least $\gamma|S|(|S|-1)/2$; in [20] a $\gamma\text{-}quasi\text{-}clique$ was defined as a maximal subset of nodes S such that the degree of each node $u\in S$ w.r.t. $G[S]$ is at least $\gamma(|S|-1)$. Due to the lack of better term that captures the generally accepted intuitive notion of community as being very dense (similar to a clique) we retained the term near-clique in Definition 3.

Next, we list some basic properties of $\alpha\text{-}near\text{-}cliques$ which may be easily proved:

Proposition 1. The following properties hold:

1. If a subset of nodes S of a graph G forms an $\alpha\text{-}near\text{-}clique$ then it also forms an $\alpha'\text{-}near\text{-}clique$ for all $\alpha'<\alpha$.
2. If a subset of nodes S of a graph G forms a $1\text{-}near\text{-}clique$ then the induced subgraph $G[S]$ is a clique.
3. If the set of nodes of a graph G forms an $\alpha\text{-}near\text{-}clique$ then $C(G)\geq\alpha$.

The parameter α in Definitions 2 and 3 serves to quantify the strength of relationship between the nodes that make up a

community: (1) If $\alpha = 0$, then any subset of nodes would constitute an α-*alliance*; On the other hand, if $\alpha = 1$, then an α-*alliance* is the same as a defensive alliance; (2) If $\alpha = 0$, then any subset of nodes constitutes an α-*near-clique*; If $\alpha = 1$, then only the nodes of a clique would satisfy the definition of an α-*near-clique* (Proposition 1).

III. FRAMEWORK FOR MINING THE PARAMETERS THAT CHARACTERIZE COMMUNITIES

The framework described below is inspired by the work in [12] and [18], where a similar approach was followed to characterize the *correlation profile* of the Internet and to discover *network motifs* (small subgraphs that serve as building blocks in some web-like networks), respectively.

A. Description of the Framework

Consider a graph G representing a real, web-like network and a subgraph of type T_0 given as a candidate definition of community. The proposed framework consists of the following three major steps:

1. Compute the concentration of subgraphs of type T_0 in the graph G;
2. Select a suitable random graph G_{rand} and compute the concentration of subgraphs of type T_0 in that graph;
3. Compare the concentrations found in Steps 1 and 2 to determine the statistical significance of the concentration of subgraphs of type T_0 in the graph G.

The key feature of the proposed framework is the use of the statistical significance of the concentration of subgraphs of type T_0 as a measure of the suitability of T_0 in defining community. All three steps listed above may be implemented in various different ways. In the rest of the paper we discuss a particular implementation of this framework, some of the challenges that arise while implementing it, and a set of experiments for studying the suitability of alliance and near-clique in defining the communities of some real web-like networks.

B. Estimating the Concentration of Communities by Sampling

To estimate the concentration of alliances and near-cliques we have employed a refined version of a sampling algorithm proposed in [12]. The most compelling reason for choosing this algorithm was that it computes an *unbiased* estimate of subgraph concentration which *converges quickly* to the true concentration.

1) Description of the Sampling Algorithm

The pseudo-code of the sampling algorithm proposed in [12] is shown in Algorithm 1, below. The procedure `SubgraphConcentration` generates $NSamples$ random samples (subgraphs) of size $SampleSize$ from a given graph $G(V,E)$, and uses them to derive an estimate for the concentration of subgraphs of a given type $SubgrType$ and size $SampleSize$ in the graph G.

Algorithm 1: **`SubgraphConcentration`**

input: G : an undirected graph
SampleSize : sample size; an integer
NSamples : number of samples; an integer
SubgrType : the type of subgraph being sampled
output: an estimate for the concentration of subgraphs of type *SubgrType* in G

1. **real:** $W_{SubgrType}$, W_{Total}, P
2. **graph:** $G_{SampleSize}$
3. $W_{SubgrType} = W_{Total} = 0$
4. **for** $i = 1$ **to** *NSamples* **do**
5. $G_{SampleSize}$ = `GenerateRandSample`(G, *SampleSize*)
6. P = `GetProbSample`(G, *SampleSize*, $G_{SampleSize}$)
7. $W_{Total} = W_{Total} + 1/P$
8. **if** ($G_{SampleSize}$ is of type *SubgrType*) **then**
9. $W_{SubgrType} = W_{SubgrType} + 1/P$
10. **return** $W_{SubgrType} / W_{total}$

Key to the procedure `SubgraphConcentration` are the functions `GenerateRandSample` which produces a random sample, and `GetProbSample` which computes the probability that `GenerateRandSample` will produce a given fixed sample.

The function `GenerateRandSample` begins by constructing a sequence of trees $\{T_i(V_i, E_i)\}_{i=1,\ldots,SampleSize}$ as follows: (i) the tree T_1 consists of an edge e_1 selected uniformly at random (u.a.r.) from G; (ii) for each $i = 2,\ldots,SampleSize$ the tree T_i is given by $T_i = T_{i-1} \cup \{e_i\}$ where e_i is selected u.a.r. among those edges in $E \setminus E_{i-1}$ which are incident on V_{i-1}. The random sample produced by `GenerateRandSample` is then $G\left[V_{SampleSize}\right]$—the subgraph induced by the set of nodes of the final tree $T_{SampleSize}$.

Given a fixed tree $T_{SampleSize}$ generated as described above, the function `GetProbSample` computes the probability P that `GenerateRandSample` will produce the tree $T_{SampleSize}$. The probability P is obtained by summing up over all permutations σ of the edges of $T_{SampleSize}$ the individual probabilities P_σ—where P_σ denotes the probability of the tree $T_{SampleSize}$ being generated in the sequence specified by the permutation σ.

Finally, the *estimator* of the concentration of subgraphs of type $SubgrType$ in G is given by $W_{SubgrType} / W_{Total}$ where W_{Total} is the sum of the weights $1/P$ over all samples, and $W_{SubgrType}$ is the sum of the weights $1/P$ over all samples that are of type $SubgrType$.

2) Proposed Enhancement of the Sampling Algorithm

For a given tree $T \subset G$, only a fraction of the permutations of its edges will specify sequences in which it is possible to generate this tree. Let us call a permutation σ of the set of edges of T *feasible* if the sampling procedure may generate this tree in the sequence specified by σ. Otherwise, let us call this permutation *infeasible*. As an example, consider the tree formed by the edges $e_1 = (3,5)$, $e_2 = (5,6)$, and $e_3 = (6,8)$ in the graph shown in Fig. 1. Among the six possible permutations of these three edges, two are infeasible ($\{e_1, e_3, e_2\}$ and $\{e_3, e_1, e_2\}$) while the remaining four are feasible.

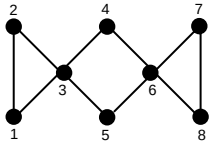


Fig. 1. A simple graph on 8 nodes.

It may be seen that a permutation σ of the set of edges $\{e_1, \ldots, e_{SampleSize-1}\}$ of a tree of size $SampleSize$ is feasible if and only if the subgraph induced by the edges $e_{\sigma(1)}, \ldots, e_{\sigma(j)}$ is *connected* for all $j = 1, \ldots, SampleSize - 1$.

The proposed speedup of the sampling algorithm, which is a natural consequence of the preceding arguments, consists in modifying the function `GetProbSample` so that it does not iterate over all permutations of the set of edges of a sampled tree, but only *over all feasible* permutations. To implement this change we devised an efficient procedure for enumerating all feasible permutations of a tree, and integrated it with the original sampling algorithm. How much is the performance of the procedure `GetProbSample` improved by applying this change? To answer this question, first we make the following observations:

Observation 1. The worst-case input to `GetProbSample` is a *star*, i.e., a tree of diameter one consisting of a node u to which all other nodes are adjacent. Every permutation of the edges of this tree is feasible and thus the number of feasible permutations in this case is $(S-1)!$.

Observation 2. The best-case input to `GetProbSample` is a *path* of length $S-1$. In this case it is easy to see that the number of feasible permutations is 2^{S-2}.

In general, shallow trees have more feasible permutations than the deeper ones. To estimate the average number of feasible permutations for a sample produced by `GenerateRandSample`, we employed simulation over a real web-like network: the Free Online Dictionary of Computing Terms (FOLDOC) (described in Section IV). Table II shows for each sample size between 5 and 10 the average (3rd column), the smallest (2nd column), and the largest (4th column) number of feasible permutations in a sample of that size. For a fixed sample size, the mean number of feasible permutations was computed by averaging over 1000 samples taken from FOLDOC, while the smallest and the largest numbers were calculated through the expressions given in Observations 1 and 2, respectively. Table II shows clearly that the average case is much closer to the best case than it is to the worst case.

TABLE II
NUMBER OF FEASIBLE PERMUTATIONS

SampleSize	Best Case	Average Case	Worst Case
5	8	14	24
6	16	47	120
7	32	185	720
8	64	1041	5040
9	128	5397	40320
10	256	43330	362880

C. Choosing the Baseline Random Graph

How can we choose a "suitable" random graph which may be used as a baseline for deriving the statistical significance of subgraph concentration? The first candidate that comes to mind is, of course, the classical Erdös-Rényi random graph $G_{n,p}$. The problem with $G_{n,p}$ is that the dense-subgraph concentration in it is too small to allow a comparison with the concentration in real, web-like networks. As an illustration, consider the expected number of triangles in $G_{n,p}$. Using a technique of the theory of random graphs, this expectation may be derived as follows: (i) Fix three nodes u, v, w in $G_{n,p}$. The probability that these three nodes induce a triangle is p^3 since each potential edge of $G_{n,p}$ is present with probability p, independently of the other edges. (ii) There are $\binom{n}{3}$ distinct ways of choosing triples of nodes u, v, w. Hence, the expected number of triangles is $\binom{n}{3} p^3$. Now, consider the FOLDOC network (Section IV) which has $n = 13,356$ nodes and $m = 91,465$ edges. In a classical random graph with the same number of nodes and edges, the edge density would be $p \approx 2m/[n(n-1)] = 3.2 \times 10^{-6}$. Thus, the expected number of triangles in this random graph would be approximately 3×10^{-3}, i.e., less than one. Similar results can be obtained for cliques of bigger size and near-cliques.

To avoid this issue, instead of $G_{n,p}$ we used as a baseline a *random graph with given degree sequence*. The degree sequence of this random graph, which we denote by G_{rand}, is extracted from the real-network under investigation. Since it is difficult to generate a random graph with a given degree sequence directly, we worked with the related *configuration* model proposed in [2]: To generate a random configuration with n nodes on the degree sequence $\mathcal{D} = (d_1, \ldots, d_n)$, the following two steps are carried out: (1) Create a set L containing d_i copies of node i for $i = 1, \ldots, n$; (2) Choose a random matching of the elements of L. Each configuration represents an underlying *multigraph* (i.e., a graph that might contain self-loops and parallel edges) whose edges are defined by the pairs in this matching.

D. *Determining the Statistical Significance of Subgraph Concentration*

We follow [12] in using the z-score defined by:

$$z = \frac{C_{real} - \mathbb{E}(C_{rand})}{\mathrm{STD}(C_{rand})} \quad (1)$$

as a measure of the statistical significance of the concentration. In the expression above, $\mathbb{E}(C_{rand})$ denotes the expected value, while $\mathrm{STD}(C_{rand})$ denotes the standard deviation of the concentration of a subgraph in the random graph G_{rand}.

IV. EXPERIMENTAL RESULTS

We have conducted several experiments on a large web-like network: the Free OnLine Dictionary Of Computing Terms (FOLDOC), which is a searchable dictionary of terms related to computing. The dataset for this network was downloaded from `www.pajek.com`. The nodes in the graph representation of FOLDOC represent computing terms; a directed arc (u,v) means that the term v is used to describe the meaning of term u. First, we converted the original *directed* graph into an undirected one by ignoring the orientation of arcs. Then, we converted the resulting undirected graph into a simple one by replacing each group of parallel edges with a single edge. Table III shows some parameters of this simplified version of FOLDOC.

TABLE III
SOME PARAMETERS OF FOLDOC NETWORK

Parameter		Value	
n		13356	
m		91465	
min. degree		2	
max. degree		728	
avg. degree		13.697	
avg. distance		5.85	
γ		3.0	
C(*G*)		0.3379	

A. Concentration of Alliances and Near-cliques

Employing the enhanced sampling algorithm described earlier, we determined the concentration of α-$alliances$ and α-$near$-$cliques$ in FOLDOC, for several distinct values of sample size.

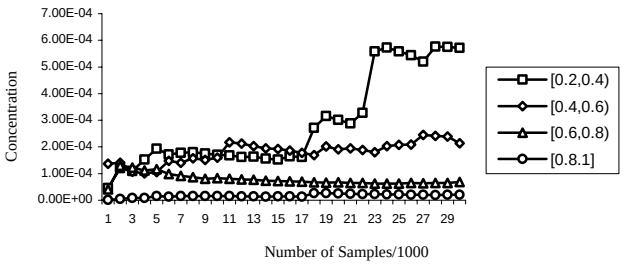


Fig. 2. Concentration of near-cliques.

Figure 2 shows the concentration of α-$near$-$cliques$ for four different ranges of parameter α: $(0.2,0.4],\ldots,(0.8,1]$. In each range, the concentration was computed for each number of samples between 1000 and 30000 in increments of 1000.

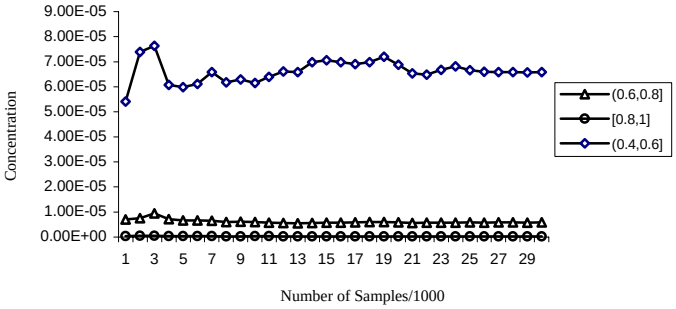


Fig. 3. Concentration of alliances.

Figure 3 shows the concentration of α-$alliances$ for three different ranges of parameter α: $[0.4,0.6),(0.6,0.8],(0.8,1]$. Again, the computation was carried out thirty times—once after each thousands of samples, up to 30000.

Figures 2 and 3 clearly show that after 30000 samples the concentration values converge for both alliances and near-cliques. Our sampling experiments indicated that, in general, the concentration of near-cliques was higher than that of alliances. As an illustration, in Table IV, we have shown some concentration values for near-cliques and alliances of size six for various ranges (buckets) of parameter α. As seen from this table, in each range, the concentration of near-cliques is about 1000 times higher than that of alliances.

TABLE IV
CONCENTRATION OF ALLIANCES AND NEAR-CLIQUES

Bucket	Near-cliques	Alliances
(0.2, 0.4]	0.179851	0.00150029
(0.4, 0.6]	0.038481	6.58933e-05
(0.6, 0.8]	0.00901258	5.85393e-06
(0.8, 1]	0.00140559	3.08555e-07

B. Statistical Significance

Table V shows the z-scores of the concentration of near-cliques and alliances for different ranges of parameter α. These scores were computed through equation (1). The $\mathbb{E}(C_{rand})$ and $\mathrm{STD}(C_{rand})$ were computed through sampling in the random graph G_{rand} which has the same degree sequence as FOLDOC.

TABLE V
Z-SCORES OF ALLIANCES AND NEAR-CLIQUES

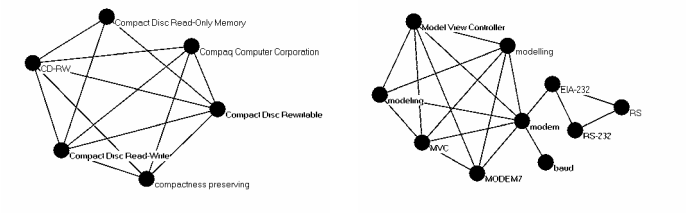
Bucket	Near-cliques	Alliances
(0.2, 0.4]	47	33
(0.4, 0.6]	136	52
(0.6, 0.8]	86	38
(0.8, 1]	22	18

The results shown in Table V, indicate that the statistical significance of the concentration of near-cliques is higher than that of alliances. Of particular interest, are the z-scores that

correspond to large values of parameter α, which may be interpreted as the statistical significance of "strong" communities. Table V, shows that near-cliques in the ranges $(0.4, 0.6]$ or $(0.6, 0.8]$, which are expected to represent groups of strongly-related nodes, have a very significant concentration in FOLDOC.

C. Highly Clustered Subgraphs

In order to examine the degree to which "strong" communities are made up of related computing terms, we used the Pajek network visualization tool (available at `www.pajek.com`) to draw several randomly chosen $\alpha\text{-}near\text{-}cliques$ with large values of parameter α. Figures 4 shows two such subgraphs which are representative of the ones that were visually inspected.



(a) (b)

Fig. 4. (a) A subgraph of FOLDOC with six nodes, in which the minimum clustering coefficient is 0.5 and average clustering coefficient is 0.8. (b) A subgraphof FOLDOC with ten nodes in which the average clustering coefficient is 0.7.

In the first example (Fig. 4(a)) it may be seen that all the terms except one are related to "compact disk". In the second example (Fig. 4(b)) the terms seem to belong to two different groups: one about "modem" and another about "model".

V. Conclusion

We proposed a new framework for evaluating the suitability of various graph theoretic definitions of community. Several issues remain as topics of our research: First, it would be desirable to modify the discussed sampling algorithm so that larger, disk-resident graphs may be investigated. Second, the proposed framework should be applied to other web-like networks to have a better understanding of the effectiveness of near-cliques and alliances in defining community. Finally, it is important to devise a formal method for tying the statistical significance of concentration with the theme (or, function) of nodes in a community.

References

[1] J. Abello, M. G. C. Resende, and S. Sudarsky, "Massive Quasi-Clique Detection," in *Proceedings of the 5th Latin American Symposium on Theoretical Informatics*: Springer-Verlag, 2002, pp. 598-612.

[2] E. A. Bender and E. R. Canfield, "The asymptotic number of labeled graphs with given degree sequences," *J. Combinatorial Theory Ser. A*, vol. 24(3), pp. 296-307, 1978.

[3] A. Cami, H. Balakrishnan, N. Deo, and R. D. Dutton, "On the complexity of finding optimal global alliances," to appear in *J. Comb. Math. Comb. Comp.*

[4] N. Deo and A. Cami, "A greedy community-mining algorithm based on clustering coefficient," presented at 36th Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Boca Raton, FL, May 2005 (to appear in *Congressus Numerantium*, vol. 168).

[5] J.-P. Eckmann and E. Moses, "Curvature of co-links uncovers hidden thematic layers in the World Wide Web," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 5825-5829, 2002.

[6] C. Faloutsos, K. S. McCurley, and A. Tomkins, "Fast discovery of connection subgraphs," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. Seattle, WA, USA: ACM Press, 2004, pp. 118-127.

[7] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient identification of Web communities," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. Boston, Massachusetts, United States: ACM Press, 2000, pp. 150-160.

[8] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-Organization and Identification of Web Communities," *Computer*, vol. 35, pp. 66-71, 2002.

[9] D. Gibson, R. Kumar, and A. Tomkins, "Discovering large dense subgraphs in massive graphs," in *Proceedings of the 31st international conference on Very large data bases*. Trondheim, Norway: VLDB Endowment, 2005, pp. 721-732.

[10] G. Jeh and J. Widom, "Mining the space of graph properties," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. Seattle, WA, USA: ACM Press, 2004, pp. 187-196.

[11] R. Jin, C. Wang, D. Polshakov, S. Parthasarathy, and G. Agrawal, "Discovering frequent topological structures from graph datasets," in *Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*. Chicago, Illinois, USA: ACM Press, 2005, pp. 606-611.

[12] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics (Oxford, England)*, vol. 20, pp. 1746-1758, 2004.

[13] P. Kristiansen, S. M. Hedetniemi, and S. T. Hedetniemi, "Alliances in graphs," *J. Combin. Math. Combin. Comput.*, vol. 48, pp. 157-177, 2004.

[14] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the Web for emerging cyber-communities," *Computer Networks*, vol. 31, pp. 1481, 1999.

[15] M. Kuramochi and G. Karypis, "Frequent Subgraph Discovery," *Data Mining and Knowledge Discovery*, vol. 11, pp. 243-271, 2005.

[16] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005)*, 2005, pp. 177-187.

[17] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*. New Orleans, LA, USA: ACM Press, 2003, pp. 556-559.

[18] S. Maslov, K. Sneppen, and A. Zaliznyak, "Detection of topological patterns in complex networks: correlation profile of the internet," *Physica A: Statistical and Theoretical Physics*, vol. 333, pp. 529-540, 2004.

[19] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167-256, 2003.

[20] J. Pei, D. Jiang, and A. Zhang, "On mining cross-graph quasi-cliques," in *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. Chicago, Illinois, USA: ACM Press, 2005, pp. 228-238.

[21] C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi, "Scalable mining of large disk-based graph databases," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. Seattle, WA, USA: ACM Press, 2004, pp. 316-325.

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

[23] M. J. Zaki, "Efficiently mining frequent trees in a forest," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. Edmonton, Alberta, Canada: ACM Press, 2002, pp. 71-80.