Extraction and Classification of Dense Implicit Communities in the Web Graph

Yon Dourisboure
LIUPPA - Université de Pau et des Pays de l'Adour, France
and
Filippo Geraci
Istituto di Informatica e Telematica - CNR, Italy
and
Marco Pellegrini
Istituto di Informatica e Telematica - CNR, Italy

The World Wide Web (WWW) is rapidly becoming important for society as a medium for sharing data, information and services, and there is a growing interest in tools for understanding collective behaviors and emerging phenomena in the WWW. In this paper we focus on the problem of searching and classifying *communities* in the web. Loosely speaking a community is a group of pages related to a common interest. More formally communities have been associated in the computer science literature with the existence of a locally dense sub-graph of the web-graph (where web pages are nodes and hyper-links are arcs of the web-graph). The core of our contribution is a new scalable algorithm for finding relatively dense subgraphs in massive graphs. We apply our algorithm on web-graphs built on three publicly available large crawls of the web (with raw sizes up to 120M nodes and 1G arcs). The effectiveness of our algorithm in finding dense subgraphs is demonstrated experimentally by embedding artificial communities in the web-graph and counting how many of these are blindly found. Effectiveness increases with the size and density of the communities: it is close to 100% for communities of a thirty nodes or more (even at low density). It is still about 80% even for communities of twenty nodes with density over 50% of the arcs present. At the lower extremes the algorithm catches 35% of dense communities made of ten nodes. We also develop some sufficient conditions for the detection of a community under some local graph models and not-too-restrictive hypotheses. We complete our *Community Watch* system by clustering the communities found in the web-graph into homogeneous groups by topic and labeling each group by representative keywords.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on Discrete Structures; H.2.8 [**Database Applications**]: Data Mining; H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Web graph, Communities, Dense Subgraph.

Work partially supported by the EU Research and Training Network COMBSTRU (HPRN-CT-2002-00278) and by the Italian Registry of ccTLD "it". Extended version of the paper "Extraction and Classification of Dense Communities in the Web" appeared in Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 461-470 [Dourisboure et al. 2007].

1. INTRODUCTION

Searching for social structures in the World Wide Web has emerged as one of the foremost research problems related to the breathtaking expansion of the World Wide Web. Thus there is a keen academic as well as industrial interest in developing efficient algorithms for collecting, storing and analyzing the pattern of pages and hyper-links that form the World Wide Web, since the pioneering work of Gibson, Kleinberg and Raghavan [Gibson et al. 1998]. Nowadays many communities of the real world that want to have a major impact and recognition are represented in the Web. Thus the detection of *cyber-communities*, i.e. set of sites and pages sharing a common interest, improves also our knowledge of the world in general.

1.1 Cyber-communities as dense subgraphs of the web graph

The most popular way of defining cyber-communities is based on the interpretation of WWW *hyperlinks* as *social* links [Chakrabarti et al. 1999]. For example, the web page of a conference contains an hyper-link to all of its sponsors, similarly the home-page of a car lover contains links to all famous car manufactures. In this way, the Web is modelled by the *web graph*, a *directed graph* in which each vertex represents a web-page and each arc represents an hyper-link between the two corresponding pages. Intuitively, cyber-communities correspond to dense subgraphs of the web graph.

1.1.1 *An open problem.* Monika Henzinger in a recent survey on algorithmic challenges in web search engines [Henzinger 2002] remarks that the Trawling algorithm of Kumar et al. [Kumar et al. 1999b] is able to enumerate dense bipartite graphs in the order of tens of nodes and states this open problem: "In order to more completely capture these cyber-communities, it would be interesting to detect much larger bipartite subgraphs, in the order of hundreds or thousands of nodes. They do not need to be complete, but should be dense, i.e. they should contain at least a constant fraction of the corresponding complete bipartite subgraphs. Are there efficient algorithms to detect them? And can these algorithms be implemented efficiently if only a small part of the graph fits in main memory?"

1.1.2 *Theoretical results.* From a theoretical point of view, the *dense k-subgraph* problem, i.e. finding the densest subgraph with k vertices in a given graph, is clearly NP-Hard (it is easy to see by a reduction from the max-clique problem). Some approximation algorithms with a non constant approximation factor can be found in the literature for example in [Han et al. 2000; Feige et al. 2001; Feige and Langberg 2001], none of which seem to be of practical applicability. Studies about the inherent complexity of the problem of obtaining a constant factor approximation algorithm are reported in [Hastad 1999] and [Feige 2002].

1.1.3 *Some heuristic methods.* In the literature there are a few heuristic methods to extract communities from the web (or from large graphs in general). The most important and ground breaking algorithm is due to Kumar et al. in [Kumar et al. 1999b] where the authors aim at enumerating complete bipartite subgraphs with very few vertices, then extend them to *dense bipartite subgraphs* by using local searches (based on the HITS ranking algorithm). The technique in [Kumar et al. 1999b] is aimed at detecting small complete bipartite communities, of the order of ten vertices, while the subsequent community expansion guided by the hub and authority scores of the HITS algorithm (regardless of further density considerations). In [Flake et al. 2002] Flake, Lawrence, Giles and Coetzee use the notion of

maximum flow to extract communities, but they are also limited to communities for which an initial seed node is available. In [Gibson et al. 2005] Gibson, Kumar and Tomkins use a new sampling method (shingling) based on the notion of min-wise independent permutations, introduced in [Broder et al. 2000], to evaluate the similarity of neighborhoods of vertices and then extract very large and very dense subgraphs of the web-host graph. This technique is specifically aimed to detecting very large and dense subgraphs, in a graph, like the web-host-graph of quite large average degree. The authors in [Gibson et al. 2005, Section 4.2] remark that (with a reasonable set of parameters) the shingling method is effective for dense subgraphs of over 50 nodes but breaks down below 24 nodes. Thus there is room for improvements via alternative approaches.

1.2 Our contribution

In this paper we propose two new simple characterization of dense subgraphs. From these characterization we derive a new heuristic, which is based on a two-step filtering approach. In the first filtering step we estimate efficiently the average degree and the similarity of neighbor sets of vertices of a candidate community. This initial filtering is very efficient since it is based only on degree-counting. The second filtering step is based on an iterative refinement of the candidate community aimed at removing small degree vertices (relative to the target average density), and thus increasing the average degree of the remaining core community. We test our algorithm on very large snapshots of the web graph (both for the global web-graph and for some large national domains) and we give experimental evidence of the effectiveness of the method. We have coupled the community extraction algorithm with a clustering tool that groups the communities found into homogeneous groups by topic and provide a useful user interface for exploring the community data. The user interface of the *Community Watch* system is publicly available at `http://comwatch.iit.cnr.it`. To the best of our knowledge this is the first publicly available tool to visualize cyber-communities.

1.2.1 *Target size.* In our method the user supplies a target threshold t and the algorithm lists all the communities found with average degree at least t. Naturally the lower the t-value the more communities will be found and the slower the method. In our experiments our method is still effective for values of t quite close to the average degree of the web-graphs (say within a factor 2), and communities of a few tens of nodes. Our heuristic is particularly efficient for detecting communities of large and medium size, while the method in [Kumar et al. 1999b] is explicitly targeted towards communities with a small complete bipartite core-set.

1.2.2 *Final applications.* The detection of dense subgraphs of the web-graph might serve as a stepping stone towards achieving several broader goals. One possible goal is to improve the performance of critical tools in the WWW infrastructure such as crawlers, indexing and ranking components of search engines. In this case often dense subgraphs are associated with negative phenomena such as the Tightly Knit Community (TKC) effect [Lempel and Moran 2000], link-farm spamming [Gyöngyi and Garcia-Molina 2005], and data duplication (mirroring) [Bharat et al. 2000]. In this paper, following [Kumar et al. 1999c] we place instead the accent on the "positive" aspect of cyber-communities: our intent at the moment is to provide an exploratory tool capable of extracting a synthetic description of the current status and current trends in the social structure of the WWW.

1.3 Visualization of the Communities

Given a single dense community it is easy by manual inspection to gain some hint as to its general area of interest and purpose, however gaining insight on hundreds (or thousands) of communities can become a tiresome task, therefore we have coupled our dense-subgraph extraction algorithm with a visualization tool that helps in the exploratory approach. This tool is based on the efficient clustering/labelling system described in detail in [Geraci et al. 2007]. In nutshell from each community, using standard IR techniques, we extract a vector of representative words with weights related to the words frequencies (word-vector). A clustering algorithm is applied to the word-vectors and we obtain groups of communities that are homogeneous by topic, moreover a list of representative keywords for each cluster is generated so to guide the user to assess the intrinsic topic of each cluster of communities.

1.4 Mirrors and Link-farms

Information retrieval on the WWW is complicated by the phenomenon of "data replication" (mirroring) and several forms of spamming (e.g. link-farms). For mirrors, off-line detection of such structures using the techniques in [Bharat et al. 2000] implies pairwise comparisons of all (or most if some heuristic filtering is used) pairs of web-sites, which is an expensive computation. Link-farm detection implies technique borderline with those used for community detection. In our context, however, efficiency and effectiveness of the community detection algorithm are not really impaired by such borderline phenomena. For this reason we do not attempt to filter out these phenomena before applying our algorithms. Instead we envision these steps (mirror detection and link-farm detection) as a post-processing phase in our *Community Watch* system. In particular since we perform efficiently both the community detection and community clustering we can apply mirror and link-farm detection separately and independently in each cluster thus retaining the overall system scalability.

1.5 Organization of the paper

Section 2 lists previous work on finding dense subgraphs in the web graph. Section 3 introduces the precise definition of community we use. Section 4 describes the derivation of our criteria for detecting a community, and the resulting algorithm. Section 5 gives a theoretical underpinning to the proposed algorithm by listing sufficient conditions for its asymptotic correctness, and also some non-asymptotic bounds in simpler models. Section 6 describes sufficient conditions for the detection of two dense partially overlapping community. Section 7 describes the experimental validation of the proposed algorithm. Section 8 describes the communities found on three large snapshots of the web graph. Section 9 introduces the tool *Community Watch* we have developed and used to classify the communities by content. The preliminary work in [Dourisboure et al. 2007] includes the description of the algorithm and the experiments with embedded communities. In this extended journal paper we have developed the models and the sufficient conditions for the detections of the communities in Section 5 and the case of dense overlapping communities in Section 6. Moreover, in Section 8 we give an example of the application of our classification method of communities into homogeneous groups for the graph of the domain .uk in 2005.

2. PREVIOUS WORK

Given the hypertext nature of the WWW one can approach the problem of finding cyber-communities by using as main source the textual content of the web pages, the hyperlinks structure, or both. Among the methods for finding group of coherent pages based only on text content we can mention [Broder et al. 1997]. Recommendation systems usually collect information on social networks from a variety of sources (not only link structure) (e.g. [Kautz et al. 1997]). Problems of a similar nature appears in the areas of social network analysis, citation analysis and bibliometrics, where, however, given the relatively smaller data sets involved (relative to the WWW), efficiency is often not a critical issue [Newman 2003].

Since the pioneering work [Gibson et al. 1998] the prevailing trend in the Computer Science community is to use mainly the link-structure as basis of the computation. Previous literature on the problem of finding cyber-communities using link-based analysis in the web-graph can be broadly split into two large groups. In the first group are methods that need an initial seed of a community to start the process of community identification. Assuming the availability of a seed for a possible community naturally directs the computational effort in the region of the web-graph closest to the seed and suggests the use of sophisticated but computational intensive techniques, usually based of max-flow/min-cut approaches. In this category we can list the work of [Gibson et al. 1998; Flake et al. 2000; Flake et al. 2002; Imafuji and Kitsuregawa 2003; Ino et al. 2005]. The second group of algorithms does not assume any seed and aims at finding all (or most) of the communities by exploring the whole web graph. In this category falls the work of [Kumar et al. 1999b; 1999a; Reddy and Kitsuregawa 2001; Kumar et al. 2005; Gibson et al. 2005].

Certain particular artifacts in the WWW called "link farms" whose purpose is to bias search-engines pagerank-type ranking algorithms are a very particular types of "artificial" cyber-communities that are traced using techniques bordering with those used to find dense subgraphs in general. See for example [Wu and Davison 2005; Bianchini et al. 2005].

Abello et al. [Abello et al. 2002] propose a method based on local searches with random restarts to escape local minima, which is quite computational intensive. A graph representing point to point telecommunications with 53 M nodes and 170M edges is used as input. The equipment used is a multiprocessor machine of 10 200MHz processors and total 6GB RAM memory. A timing result of roughly 36 hours is reported in [Abello et al. 2002] for an experiment handling a graph obtained by removing all nodes of degree larger than 30, thus, in effect, operating on a reduced graph of 9K nodes and 320K edges. Even discounting for the difference in equipment we feel that the method in [Abello et al. 2002] would not scale well to searching for medium-density and medium-size communities in graphs as large as those we are able to handle (up to 20M nodes and 180M edges after cleaning). Girvan and Newman [Girvan and Newman 2002] define a notion of local density based on counting the number of shortest paths in a graph sharing a given edge. This notion, though powerful, entails algorithm that do not scale well to the size of the web-graph. Spectral methods described in [Capocci et al. 2004] also lack scalability (i.e. in [Capocci et al. 2004] the method is applied to graphs from psychological experiments with 10K nodes and 70K edges).

A system similar in spirit to that proposed in this paper is *Campfire* described in [Kumar et al. 1999c] which is based on the Trawling algorithm for finding the dense core, on HITS for community expansion and on an indexing structure of

community keywords that can be queried by the user. Our system is different from Campfire first of all in the algorithms used to detect communities but also in the final user interface: we provide a clustering/labelling interface that is suitable to giving a global view of the available data.

3. PRELIMINARIES

3.1 Notions and notation

A *directed graph* $G = (V, E)$ consists of a set V of *vertices* and a set E of *arcs*, where an arc is an ordered pair of vertices. The *web graph* is the directed graph representing the Web: vertices are pages and arcs are hyperlinks.

Let u, v be any vertices of a directed graph G, if there exists an arc $a = (u, v)$, then a is an *outlink* of u, and an *inlink* of v. Moreover, v is called a *successor* of u, and u a *predecessor* of v. For every vertex u, $\mathrm{N}^+(u)$ denotes the set of its successors, and $\mathrm{N}^-(u)$ the set of its predecessors. Then, the *outdegree* and the *indegree* of u are respectively $\mathrm{d}^+(u) = |\mathrm{N}^+(u)|$ and $\mathrm{d}^-(u) = |\mathrm{N}^-(u)|$. Let X be any subset of V, the successors and the predecessors of X are respectively defined by: $\mathrm{N}^+(X) = \bigcup_{u \in X} \mathrm{N}^+(u)$ and $\mathrm{N}^-(X) = \bigcup_{u \in X} \mathrm{N}^-(u)$. Observe that $X \cap \mathrm{N}^+(X) \neq \emptyset$ is possible. A graph $G = (V, E)$ is called a *complete bipartite graph*, if V can be partitioned into two disjoint subsets X and Y, such that, for every vertex u of X, the set of successors of u is exactly Y, i.e., $\forall u \in X,\ \mathrm{N}^+(u) = Y$. Consequently for every node $v \in Y$ its predecessor set is X. Finally, let $\widetilde{\mathrm{N}}(u)$ be the set of vertices that share at least one successor with u: $\widetilde{\mathrm{N}}(u) = \{w \in V \mid \mathrm{N}^+(u) \cap \mathrm{N}^+(w) \neq \emptyset\}$.

Two more useful definitions. Define for sets A and B the relation $A \simeq_\gamma B$ when $|A \cap B| \geq \gamma |B|$, for a constant $\gamma \in [0, 1]$. Define for positive numbers a, b the relation $a \approx_\epsilon b$ when $|a - b| \leq \epsilon |a|$, for a constant $\epsilon \in [0, 1]$. When the constant can be inferred from the context the subscript is omitted.

3.2 Definitions of Web Community

The basic argument linking the (informal) notion of web communities and the (formal) notion of dense subgraphs is developed and justified in [Kumar et al. 1999b]. It is summarized in [Kumar et al. 1999b] as follows: "Web communities are characterized by dense directed bipartite subgraph". Without entering in a formal definition of density in [Kumar et al. 1999b] it is stated the hypothesis that: "A random large enough and dense enough bipartite subgraph of the Web almost surely has a core", (i.e. a complete bipartite sub-graph of size (i, j) for some small integer values, i and j). A standard definition of γ-density, as used for example in [Gibson et al. 2005], is as follows: a γ-dense bipartite subgraph of a graph $G = (V, E)$ is a disjoint pair of sets of vertices, $X, Y \subseteq V$ such that $|\{(x, y) \in E | x \in X \wedge y \in Y\}| \geq \gamma |X||Y|$, for a real parameter $\gamma \in [0 \ldots 1]$. Note that $\gamma |Y|$ is also a lower bound to the average out-degree of a node in X. Similarly a γ-dense quasi-clique is a subset $X \subset V$ such that $|\{(x, y) \in E | x \in X \wedge y \in X\}| \geq \gamma \binom{|X|}{2}$, for a real parameter $\gamma \in [0 \ldots 1]$, as in [Abello et al. 2002; Feige et al. 2001]. This notion of a core of a dense subgraph in [Kumar et al. 1999b] is consistent with the notion of γ-density for values of γ large enough, where the notion of "almost surely", (i, j)-core, "large enough", "dense enough", must be interpreted as a function of γ. Our formulation unifies the notion of a γ-dense bipartite subgraph and a γ-dense quasi-clique as a pair of not necessarily disjoint sets of vertices, $X, Y \subseteq V$ such that $\forall x \in X, |N^+(x) \cap Y| \geq \gamma |Y|$ and $\forall y \in Y, |N^-(y) \cap X| \geq \gamma' |X|$. For two constants γ

and γ' in $[0,1]$. Our definition implies that in [Gibson et al. 2005], and conversely, any γ-dense subgraph following [Gibson et al. 2005] contains a γ-dense subgraph in our definition[1].

Thus a community in the web is defined by two sets of pages, the set of the Y *centers* of the community, i.e. pages sharing a common topic, and the set X of the *fans*, i.e., pages that are interested in the topic. Typically, every fan contains a link to most of the centers, at the same time, there are few links among centers (often for commercial reasons) and among fans (fans may not know each other).

4. HEURISTIC FOR LARGE DENSE SUBGRAPHS EXTRACTION

4.1 Description

The definition of γ-dense subgraph can be used to *test* if a pair of sets $X, Y \subseteq V$ is a γ-dense subgraph (both bipartite and clique). However it cannot be used to *find* efficiently a γ-dense subgraph (X, Y) embedded in G. In the following of this section we define properties of dense sub-graphs and then we will proceed by *relaxing* them up to the point of having properties that can be computed directly on the input graph G. These properties will hold exactly (with equality) for an *isolated* complete bipartite graph (and clique), will hold approximately for an *isolated* γ-dense graph, where the measure of approximation will be related to the parameter γ. However at the end we need a final relaxation step in which we will consider the subgraphs as embedded in G.

4.1.1 *Initial intuitive outline.* First of all, let us give an initial intuition of the reason why our heuristic might work. Let $G = (V, E)$ be a sparse directed graph, and let (X, Y) be a γ-dense subgraph within G. Then, let u be any vertex of X. Since (X, Y) is a γ-dense subgraph by definition we have $\forall u \in X, \mathrm{N}^+(u) \simeq_\gamma Y$, and symmetrically $\forall v \in Y, \mathrm{N}^-(v) \simeq_{\gamma'} X$. For values $\gamma > 0.5$ the pigeon hole principle ensures that any two nodes u and v of X always share a successor in Y, thus $X \subseteq \widetilde{\mathrm{N}}(u)$, and, if every vertex of Y has at least a predecessor in X, also $Y \subseteq N^+(\widetilde{\mathrm{N}}(u))$. The main idea now is to estimate quickly, for every vertex u of G, the degree of similarity of $\mathrm{N}^+(u)$ and $\mathrm{N}^+(\widetilde{\mathrm{N}}(u))$. In the case of an isolated complete bipartite graph $\mathrm{N}^+(u) = Y$, and $\mathrm{N}^+(\widetilde{\mathrm{N}}(u)) = Y$. For an isolated γ-dense bipartite graph, we have $\mathrm{N}^+(u) \simeq_\gamma Y$ and $\mathrm{N}^+(\widetilde{\mathrm{N}}(u)) = Y$. The conjecture is that when the γ-dense bipartite graph is a subgraph of G, and thus we have the weaker relationship $Y \subseteq N^+(\widetilde{\mathrm{N}}(u))$, the excess $N^+(\widetilde{\mathrm{N}}(u)) \setminus Y$ is small compared to Y so to make the comparison of the two sets still significant for detecting the presence of a dense subgraph. We will make these concepts more precise in Section 5 and derive the main formal result of theorem 1 stating sufficient condition for the convergence of the main function we use in detecting the presence of a dense subgraph.

4.1.2 *The isolated complete case.* To gain in efficiency, instead of evaluating the similarity of successor set, we will estimate the similarity of out-degrees by counting. In a complete bipartite graph (X, Y), we have that $\forall u \in X,\ N^+(u) = Y$, therefore, $\forall u, v \in X,\ N^+(u) = N^+(v)$. The set of vertices sharing a successor with u is $\widetilde{\mathrm{N}}(u) = X$, and moreover $N^+(\widetilde{\mathrm{N}}(u)) = Y$. Passing from relations among sets

[1] It is sufficient to eliminate nodes of X of outdegree smaller than $\gamma|Y|$, and from Y those of indegree smaller than $\gamma'|X|$.

to relations among cardinalities we have that: $\forall u, v \in X,\ d^+(u) = d^+(v)$, and the degree of any node coincide with the average out-degree:

$$\mathrm{d}^+(u) = \frac{1}{|\widetilde{\mathrm{N}}(u)|} \sum_{v \in \widetilde{\mathrm{N}}(u)} \mathrm{d}^+(v).$$

4.1.3 *The isolated γ-dense case.* In a γ-dense bipartite graph, we still have $\widetilde{\mathrm{N}}(u) = X$ but now, $|Y| \geq \mathrm{d}^+(v) \geq \gamma|Y|$ for every $v \in X$. Thus we can conclude that

$$|\mathrm{d}^+(u) - \frac{1}{|\widetilde{\mathrm{N}}(u)|} \sum_{v \in \widetilde{\mathrm{N}}(u)} \mathrm{d}^+(v)| \leq (1-\gamma)|Y| \leq \frac{1-\gamma}{\gamma}\mathrm{d}^+(u).$$

For $\gamma \to 1$ the difference tends to zero. Finally assuming that for a γ-dense bipartite subgraph of G the excesses $\widetilde{\mathrm{N}}(u) \setminus X$ and $N^+(\widetilde{\mathrm{N}}(u)) \setminus Y$ give a small contribution, we can still use the above test as evidence of the presence of a dense sub-graph. At this point we pause, we state our first criterion and we subject it to criticism in order to improve it.

CRITERION 1. *If $\mathrm{d}^+(u)$ and $|\widetilde{\mathrm{N}}(u)|$ are big enough and*

$$\mathrm{d}^+(u) \approx \frac{1}{|\widetilde{\mathrm{N}}(u)|} \sum_{v \in \widetilde{\mathrm{N}}(u)} \mathrm{d}^+(v),$$

then $\left(\widetilde{\mathrm{N}}(u), \mathrm{N}^+(\widetilde{\mathrm{N}}(u))\right)$ might contain a community.

4.1.4 *A critique of Criterion 1.* Unfortunately, this criterion 1 cannot be used yet in this form. One reason is that computing $\widetilde{\mathrm{N}}(u)$ for every vertex u of big enough outdegree in the web graph G is not scalable. Moreover, the criterion is not robust enough w.r.t. noise from the graph. Assume that the situation depicted in figure 1 occurs: $u \in X$, (X, Y) induces a complete bipartite graph with $|Z| = |X| = |Y| = x$, and each vertex of Y has one more predecessor of degree 1 in Z. Then, $\widetilde{\mathrm{N}}(u) = X \cup Z$, so $\frac{1}{|\widetilde{\mathrm{N}}(u)|} \sum_{v \in \widetilde{\mathrm{N}}(u)} \mathrm{d}^+(v) = \frac{x+1}{2}$ that is far from $\mathrm{d}^+(u) = x$, so (X, Y) will not be detected.

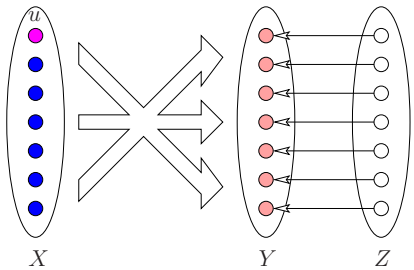


Fig. 1. A complete bipartite subgraph with $|X| = |Y| = x$, and some "noise" Z, with $|Z| = x$.

4.1.5 *Overcoming the drawbacks of Criterion 1.* Because of the shortcomings of Criterion 1 we describe a second criterion that is more complex to derive but computationally more effective and robust. As before we will start with the case of the isolated complete bipartite graph. Consider a node $u \in X$, clearly $\mathrm{N}^+(u) = Y$, and $\forall y \in \mathrm{N}^+(u)$, $\mathrm{N}^-(y) = X$, thus $\forall w \in \mathrm{N}^-(y)$, $\mathrm{N}^+(w) = Y$. Turning to the cardinalities: for a node $u \in X$, $\forall y \in \mathrm{N}^+(u)$, $\forall w \in \mathrm{N}^-(y)$ $\mathrm{d}^+(w) = |Y|$. Thus also the average value of all out-degrees for nodes in $\mathrm{N}^-(y)$ is $|Y|$. In formulae: given $u \in X$, $\forall y \in \mathrm{N}^+(u)$,

$$\frac{1}{\mathrm{d}^-(y)} \sum_{w \in \mathrm{N}^-(y)} \mathrm{d}^+(w) = |Y|.$$

Next we average over all $y \in \mathrm{N}^+(u)$ by obtaining the following equation: given $u \in X$,

$$\frac{1}{\sum_{y \in \mathrm{N}^+(u)} \mathrm{d}^-(y)} \sum_{y \in \mathrm{N}^+(u)} \sum_{w \in \mathrm{N}^-(y)} \mathrm{d}^+(w) = |Y|.$$

Finally since $\mathrm{d}^+(u) = |Y|$ we have the equality:

$$\frac{1}{\sum_{y \in \mathrm{N}^+(u)} \mathrm{d}^-(y)} \sum_{y \in \mathrm{N}^+(u)} \sum_{w \in \mathrm{N}^-(y)} \mathrm{d}^+(w) = \mathrm{d}^+(u).$$

We restate the above equality in terms of a few function easier to manipulate. Let:

$$A(u) = \sum_{y \in \mathrm{N}^+(u)} \sum_{w \in \mathrm{N}^-(y)} \mathrm{d}^+(w), \tag{1}$$

$$B(u) = \sum_{y \in \mathrm{N}^+(u)} \mathrm{d}^-(y), \tag{2}$$

and

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)}. \tag{3}$$

The test for an isolated complete bipartite graph is equivalent to requesting that $Err(u)$ is zero. Next we relax the scenario and we apply the test to an isolated γ-dense bipartite graph. Here we require that $|Err(u)|$ is small (i.e. tends to zero) when the graph is close to be a complete bipartite graph (i.e γ and γ' tend to 1), or a clique. Consider a node $u \in X$, since $\mathrm{N}^+(u) \simeq_\gamma Y$, and for a node $v \in Y$, $\mathrm{N}^-(v) \simeq_{\gamma'} X$, we get the bounds:

$$|X||Y| \geq B(u) \geq \gamma|Y|\gamma'|X|,$$

$$|Y|^2|X| \geq A(u) \geq \gamma^2|Y|^2\gamma'|X|.$$

On the other hand $|Y| \geq \mathrm{d}^+(u) \geq \gamma|Y|$. The error function $Err(u)$ is upper bounded by $1 - \gamma^2\gamma'$. For $\gamma \to 1$ and $\gamma' \to 1$ the error tends to zero.

Thus in an approximate sense the relationship is preserved for isolated γ-dense bipartite graphs. Clearly now we will make a further relaxation by considering the sets $\mathrm{N}^+(.)$ and $\mathrm{N}^-(.)$ as referred to the overall graph G, instead of just the isolated pair (X, Y).

CRITERION 2. *If* $\mathrm{d}^+(u)$ *and* $|\widetilde{\mathrm{N}}(u)|$ *are big enough and*

$$\mathrm{d}^+(u) \approx \frac{1}{\sum_{y\in \mathrm{N}^+(u)} \mathrm{d}^-(y)} \sum_{y\in \mathrm{N}^+(u)} \sum_{w\in \mathrm{N}^-(y)} \mathrm{d}^+(w),$$

then $\left(\widetilde{\mathrm{N}}(u), \mathrm{N}^+(\widetilde{\mathrm{N}}(u))\right)$ *might contain a community.*

4.1.6 *Advantages of Criterion 2.* There are several advantages in using Criterion 2. The first advantage is that the relevant summations are defined over sets $\mathrm{N}^+(.)$ and $\mathrm{N}^-(.)$ that are encoded directly in the graphs G and G^T. We will compute $\widetilde{\mathrm{N}}(u)$ in the second phase only for vertices that are likely to belong to a community. The second advantage is that the result of the inner summation can be pre-computed stored and reused. We just need to store two tables of size n ($n = |V|$), one containing the values of $\sum_{v\in \mathrm{N}^-(w)} \mathrm{d}^+(v)$, the other containing the indegrees. Thirdly, the criterion 2 is much more robust than criterion 1 to noise, since the outdegree of every vertex of X is counted many times. For example, in the situation depicted in figure 1, we obtain the following result:

$\forall u \in X$ and $w \in \mathrm{N}^+(u)$, $\sum_{v\in \mathrm{N}^-(w)} \mathrm{d}^+(v) = x^2 + 1$.

Thus, $\forall u \in X$,
$\frac{1}{\sum_{w\in \mathrm{N}^+(u)} \mathrm{d}^-(w)} \sum_{w\in \mathrm{N}^+(u)} \sum_{v\in \mathrm{N}^-(w)} \mathrm{d}^+(v) = \frac{x(x^2+1)}{x(x+1)} \simeq x$.
A more general analysis of the effects of the "noise" from the graph G is postponed to Section 5.

4.1.7 *Final refinement step.* Finally, let u be a vertex that satisfies the criterion 2, we construct explicitly the two sets $\widetilde{\mathrm{N}}(u)$ and $\mathrm{N}^+(\widetilde{\mathrm{N}}(u))$. Then, we extract the community (X, Y) contained in $\left(\widetilde{\mathrm{N}}(u), \mathrm{N}^+(\widetilde{\mathrm{N}}(u))\right)$ thanks to an iterative loop in which we remove from $\widetilde{\mathrm{N}}(u)$ all vertices v for which $\mathrm{N}^+(v) \cap \mathrm{N}^+(\widetilde{\mathrm{N}}(u))$ is small, and we remove from $\mathrm{N}^+(\widetilde{\mathrm{N}}(u))$ all vertices w for which $\mathrm{N}^-(w) \cap \widetilde{\mathrm{N}}(u)$ is small.

4.2 Algorithms

In figures 2 and 3 we give the pseudo-code for our heuristic. Algorithm RobustDensityEstimation detects vertices that satisfy the filtering formula of criterion 2, then function ExtractCommunity computes $\widetilde{\mathrm{N}}(u)$ and $\mathrm{N}^+(\widetilde{\mathrm{N}}(u))$ and extracts the community of which u is a fan. This two algorithms are a straightforward application of the formula in the criterion 2.

4.3 Role of the auxiliary input parameters

The input parameter t is a size parameter and indicates the lower bound target to the average degree of the communities to be detected. We tested our method for values as low as $t = 8$, meaning that we search for all communities with average

```
Algorithm RobustDensityEstimation
Input: A directed graph G = (V, E), a threshold for degrees
Result: A set S of dense subgraphs detected by vertices of outdegrees > threshold
begin
    Init:
        forall u of G do
            forall v ∈ N⁻(u) do
                TabSum[u] ← TabSum[u] + d⁺(v)
            end
        end
    Search:
        forall u that is not already a fan of a community and s.t. d⁺(u) > threshold do
            sum ← 0;
            nb ← 0;
            forall v ∈ N⁺(u) do
                sum ← sum + TabSum[v];
                nb ← nb + d⁻(v);
            end
            if sum/nb ≃ d⁺(u) and nb > d⁺(u) × threshold then
                S ← S ∪ ExtractCommunity(u);
            end
        end
    Return S;
end
```

Fig. 2. RobustDensityEstimation performs the main filtering step.

```
Function ExtractCommunity
Input: A vertex u of a directed graph G = (V, E). Slackness parameter ϵ
Result: A community of which u is a fan
begin
    Initialization:
        forall v ∈ N⁺(u) do
            forall w ∈ N⁻(v) that is not already a fan of a community do
                if d⁺(w) > (1 − ϵ)d⁺(u) then mark w as potential fan
            end
        end
        forall potential fan v do
            forall w ∈ N⁺(v) do
                mark w as potential center;
            end
        end
    Iterative refinement:
        repeat
            Unmark potential fans of small local outdegree;
            Unmark potential centers of small local indegree;
        until Number of potential fans and centers have not changed significatively

    Update global data structures:
        forall potential fan v do
            forall w ∈ N⁺(v) that is also a potential center do
                TabSum[w] ← TabSum[w] − d⁺(v);
                d⁻(w) ← d⁻(w) − 1;
            end
        end
    Return (potential fans, potential centers);
end
```

Fig. 3. ExtractCommunity extracts the dense subgraph.

degree above 8. The parameter t is needed for several reasons. The formal definition of web community as a quasi-dense bipartite graph (or quasi-dense clique) has some limit not interesting cases. For example a single edge (the graph $K_{1,1}$) satisfies the definition with 100% relative density but it is not interesting in a typical application. Thus it make sense to have $t \geq 2$. For an intermediate range of values, say $t \in [2, 7]$, enumerative methods such as the Trawling algorithm are fast and are able to discover greedily all the small dense bipartite graphs. Working in the range $t \geq 8$ we are looking in a region not well covered by other techniques. A second reason for using values of t not too small is due to the mutually reinforcing disruptive effect of low density (as captured by the parameters γ, γ') and of the noise induced by the graph on a small non-isolated community. Experiments in section 7 show that at density below 75%, for a small number of nodes: $t = 10$, the background noise in the model we adopted induces always a failure of the filtering criterion. The second parameter is a bound on the acceptable relative error $Err(u)$ for the node u. In our experiments we adopted the values $\epsilon = 0.2$ and $\epsilon = 0.25$. Since the results are essentially identical we concluded that the value $\epsilon = 0.2$ is stable w.r.t. the problem of finding communities in the instances of web graph at our disposal and we report the results for $\epsilon = 0.2$ only. Using the analysis in section 4, a relative error bound ≤ 0.2 allows to detect isolated quasi-dense communities with $\gamma = \gamma' \geq 0.9$. This rough analysis is consistent with the synthetic experimental results in section 7 where, even in the presence of background error, we capture more than 80% of the artificial communities of density above 75% and size above 20 nodes. In general we observe that the experimental results in section 7 give us performances in terms of density/size that are better than the worst case predictions as derived in sections 4 and 5. This is due to two reasons. The first reason is that failure to detect a small error value at node u belonging to a community implies the co-occurrence of several different worst case effects (the effects due to the background-noise and those due to the quasi-density of the community are rather weakly correlated). Moreover it is sufficient for the criterion to hold for one of the nodes of a community to discover the whole community. This fact explains why mid-size communities (above 30 nodes) are detected in practice even with quite low density (below 50%) and in presence of background noise.

4.4 Handling of overlapping communities

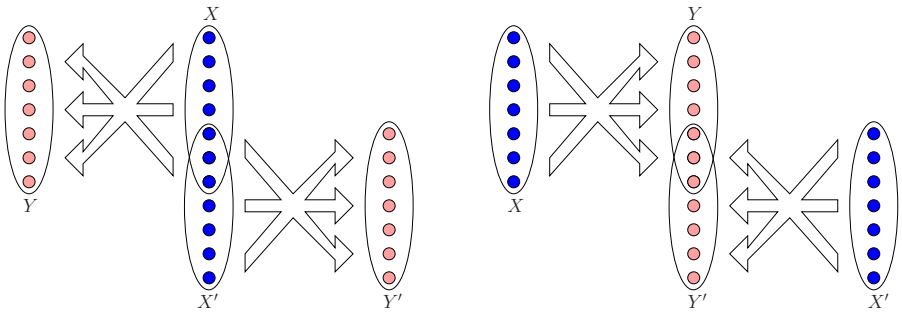
Our algorithm can capture also partially overlapping communities. This case may happen when we have older communities that are in the process of splitting or newly formed communities in the process of merging. However overlapping centers and overlapping fans are treated differently, since the algorithm is not fully symmetric in handling fans and centers.

Communities sharing fans. The case depicted in Figure 4(a) is that of overlapping fans. If the overlap $X \cap X'$ is large with respect to $X \cup X'$ then our algorithm will just return the union of the two communities $(X \cup X',\ Y \cup Y')$. Otherwise when the overlap $X \cap X'$ is not large the algorithm will return two communities: either the pairs $(X,\ Y)$ and $(X' \setminus X,\ Y')$, or the pairs $(X',\ Y')$ and $(X \setminus X',\ Y)$. So we will report both the communities having their fan-sets overlapping, but the representative fan sets will be split. The notion of large/small overlap is a complex function of the degree threshold and other parameters of the algorithm. In either case we do not miss any important structure of our data.

Communities sharing centers. Note that the behavior is different in the case of overlapping centers. A vertex can be a center of several communities. Thus, in

the case depicted in Figure 4(b), if the overlap $Y \cap Y'$ is big with respect to $Y \cup Y'$, then we will return the union of the two communities $(X \cup X',\ Y \cup Y')$, otherwise we will return exactly the two overlapping communities $(X,\ Y)$ and $(X',\ Y')$. In either case we do not miss any important structure of our data. Observe that the last loop of function ExtractCommunity removes *logically* from the graph all arcs of the current community, but not the vertices. Moreover, a vertex can be fan of a community and center of several communities. In particular it can be fan and center for the same community, so we are able to detect dense quasi bipartite subgraphs as well as quasi cliques.

A quantitative assessment of the properties of the filter in the case of overlapping communities in postponed to Section 6.



(a) Communities sharing fans (b) Communities sharing centers

Fig. 4. Two cases of community intersection

4.5 Complexity analysis

We perform now a semi-empirical complexity analysis in the standard RAM model. The graph G and its transpose G^T are assumed to be stored in main memory in such a way as to be able to access a node in time $O(1)$ and links incident to it in time $O(1)$ per link. We need $O(1)$ extra storage per node to store in-degree, out-degree, a counter `TabSum`, and a tag bit. Algorithm RobustDensityEstimation visits each edge at most once and performs $O(1)$ operations for each edge, thus has a cost $O(|V| + |E|)$, except for the cost of invocations of the ExtractCommunity function. Potentially the total time cost of the invocations of ExtractCommunity is large, however experimentally the time cost grows only linearly with the number of communities found. This behavior can be explained as follows. We measured that less than 30% of the invocations do not result in the construction of a community (see Table V), and that the inner refinement loop converges on average in less than 3 iterations (see Table IV). If the number of nodes and edges of a community found by ExtractCommunity for u is proportional by a constant to the size of the bipartite sub-graph $\left(\widetilde{\mathrm{N}}(u), \mathrm{N}^+(\widetilde{\mathrm{N}}(u))\right)$ then we are allowed to charge all operations within invocations of ExtractCommunity to the size of the output. Under these conditions each edge is charged on average a constant number of operations, thus explaining the observed overall empirical complexity $O(|V| + |E| + |Output|))$.

4.6 Scalability

The algorithm we described, including the initial cleaning steps, can be easily converted to work in the streaming model, except for procedure ExtractCommunity that seems to require the use of random access of data in core memory. Here we want to estimate with a "back of the envelope" calculation the limits of this approach using core memory. The purpose of the calculation is to establish whether machines are available with sufficient core memory to be able to handle the whole web graph. Andrei Broder et al. [Broder et al. 2000] in the year 2000 estimated the size of the indexable web graph at 200M pages and 1.5G edges (thus an average degree about 7.5 links per page, which is consistent with the average degree 8.4 of the WebBase data of 2001). A more recent estimate by Gulli and Signorini [Gulli and Signorini 2005] in 2005 gives a count of 11.5G pages. The latest index-size war ended with Google claiming an index of 25G pages. The average degree of the webgraph has been increasing recently due to the dynamic generation of pages with high degree, and some measurements give a count of 40.[2] The initial cleaning phase reduces the WebBase graph by a factor 0.17 in node count and 0.059 in the Edge count. Thus using these coefficients the cleaned web graph might have 4.25G nodes and 59G arcs. The compression techniques in [Boldi and Vigna 2004] for the WebBase dataset achieves an overall performance of 3.08 bits/edge. These coefficient applied to our cleaned web graph give a total of 22.5Gbytes to store the graph. Storing the graph G and its transpose we need to double the storage (although here some saving might be achieved), thus achieving an estimate of about 45Gbytes. Our calculation are very rough and rely on the assumption that all the conversion factors apply linearly to the whole web-graph. Each such assumption could be challenged, however, even if we are off the mark by a factor 10, and the real size is close to 450Gbytes, still we stay within feasibility. For example IBM System Z9 sells in configurations up to 64 GB of RAM core memory, while an HP 9000 Superdome Server sells in configurations up to 2TB of RAM core memory, although with a more expensive price tag.

5. ANALYSIS OF ROBUSTNESS IN PRESENCE OF NOISE

In this section we develop a deeper understanding of the formula used by the criterion 2 to detect nodes belonging to a dense community. We perform three types of analysis. In the first type of analysis we employ a simplified model for the structure of the noise graph. In particular we will assume a "sparse" model for the noise, i.e. a node in the noise component can be linked to only one of the community nodes. Next we will analyze the same simplified model in the other extreme case of a "dense" noise graph (i.e. a node in the noise component can be linked to many of the community nodes) In both cases we can derive fairly tight non-asymptotic bounds for the size of the noise component that still allows detection of the a community.

In the third case we employ a more general model taking into account all possible contributions to the noise graph. In this case exact bounds are hard to derive, but we can prove an asymptotic bound under some natural conditions.

[2] S. Vigna and P. Boldi, personal communication.

5.1 Bounds on the level of noise: the sparse case

The sparse XYZQ model is as follows. We have four sets of nodes: X, Y, Z, Q, where $|X| = |Y| = x$, $|Z| = kx$, and $|Q| = mx$. The pair (X, Y) is a complete bipartite graph, that is each element $\xi_i \in X$ has x links to elements of Y. Each element $\eta_j \in Y$ has x in-links from elements of X. Moreover each element $\xi_i \in X$ has m_i links to elements of Q, such that $\sum_i m_i = mx$ (so m is the average), and for every i, $m_i \leq |Q| = mx$. Each element of Q has a unique in-link and no out-links. Each element $\eta_j \in Y$ has k_j in-links from elements of Z, such that $\sum_j k_j = kx$ (so k is the average), and for every j, $k_j \leq |Z| = kx$. Each element of Z has a unique out-link and no in-links.

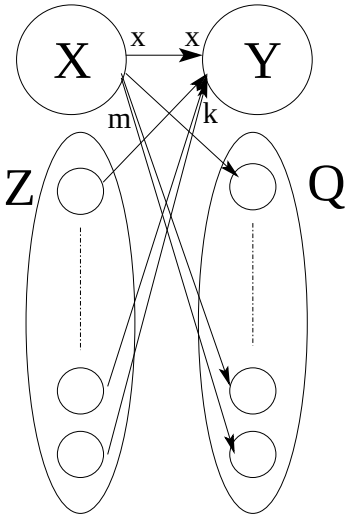


Fig. 5. The Sparse XYZQ Model. The figure indicates the 4 node classes and the 3 edge classes. Next to each edge class is indicated the symbol used in the paragraph to denote the corresponding indegree and outdegree of the nodes.

Let $Err(u)$ the relative error of formula (3) when evaluated for a given node $u \in X$. We will first give an explicit formula for $Err(u)$. Next we see several conditions that imply $|Err(u)| \leq \epsilon$ for an $\epsilon \in [0, 1]$.

Let $u = x_i$ be a node in X By an abuse of notation we indicate with m_u the noise value for u, thus $d^+(u) = x + m_u$.

We have $N^+(u) = Y \cup Q_u$ where $Q_u \subset Q$ is the set of successors of u in Q and $|Q_u| = m_u$. Let y be a generic element of $N^+(u)$. If $y \in Y$ then $N^-(y) = X \cup Z_y$ where Z_y is the set of predecessors of y in Z, and $|Z_y| = k_y$. So $d^-(y) = x + k_y$. If $y \in Q_u$ then $N^-(y) = u$, and $d^-(y) = 1$. Now we can evaluate:

$$B(u) = \sum_{y \in N^+(u)} D^-(y) = \sum_{i=1}^{x} (x + k_i) + \sum_{j=1}^{m_u} 1 = x^2 + kx + m_u$$

$$A_1(y) = \sum_{w \in N^-(y), y \in Y} D^+(w) = \sum_{i=1}^{x} (x + m_i) + k_y = x^2 + mx + k_y$$

$$A_2(y) = \sum_{w \in N^-(y), y \in Q_u} D^+(w) = \sum_{w=u} D^+(w) = x + m_u$$

$$A(u) = \sum_{y \in Y} A_1(y) + \sum_{y \in Q_u} A_2(y) = \sum_{i=1}^{x} (x^2 + mx + k_i) + \sum_{i=1}^{m_u} (x + m_u) = x^3 + mx^2 + kx + m_u^2 + m_u x$$

The relative error on node u is:

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)} = 1 - \frac{x^3 + mx^2 + kx + m_u^2 + m_u x}{(x + m_u)(x^2 + kx + m_u)}$$

Case 1. Assume $Z = \emptyset$, thus $k = 0$. We have:

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)} = 1 - \frac{x^3 + mx^2 + m_u^2 + m_u x}{(x + m_u)(x^2 + m_u)}$$

easy algebraic manipulations lead to the following statement: $|Err(u)| \leq \epsilon$ if and only if:

$$x^2 |m_u - m| \leq \epsilon (x + m_u)(x^2 + m_u). \tag{4}$$

Since m is the average of the distributions of the values for m_u, there is a node u such that $0 \leq m_u \leq m$. When $m_u = 0$ the condition reduces to the inequality: $m \leq \epsilon x$. When $m_u = m$ the left side of the inequality is null and the condition is always satisfied. In general we may observe that the left hand side of 4 is minimized by the node u having the value m_i that is the closest-to-mean. Although an a adversary can build degree distributions in which the gap between mean value and the closest-to-mean is close to m, when the series of m_i is drawn randomly from a smooth distribution, then we have an high probability of drawing values close to the mean. In this case we can handle easily a large noise count mx.

Case 2. Assume $Q = \emptyset$, thus $m = 0$, and $m_u = 0$ for all u. Thus

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)} = 1 - \frac{x^3 + kx}{x(x^2 + kx)}$$

easy algebraic manipulations lead to the following statement: $|Err(u)| \leq \epsilon$ if and only if:

$$k(1 - (1/x) - \epsilon) \leq \epsilon x \tag{5}$$

This inequality is satisfied when $k \leq \epsilon x$.

Case 3. Assume $Q \neq \emptyset$ and $Z \neq \emptyset$. In general we have that $|Err(u)| \leq \epsilon$ if and only if:

$$|x^2(k + m_u - m) + xk(m_u - 1)| \leq \epsilon (x + m_u)(x^2 + kx + m_u) \tag{6}$$

If $m_u = 0$ for some node u, we have the condition: $|x(k - m) - k| \leq \epsilon (x^2 + k)$. Note that the left hand side of this inequality is minimized when $m - k = k/x$, thus

the effect of the noise from the two sets Q and Z does cancel out when the average noise from each set has a comparable mean value. Qualitatively, the interference of the two error sources has a canceling effect.

If $m_u \neq 0$, the leading term as a polynomial in x of the left hand side of the above inequality is smaller than the leading term of the right hand side when $k - m + m_u \leq \epsilon x$, thus we can still detect the community when the average noise is a fraction of x, since there must exist a node u with $m_u \leq m$.

5.2 Bounds on the level of the noise: the dense case

Consider the following model with sets X, Y, Z, Q, where $|X| = |Y| = x$, $|Z| = x$ and $|Q| = x$. The pair (X, Y) is a complete bipartite graph, each element $\xi_i \in X$ has x links to elements of Y. Each element $\eta_j \in Y$ has x in-links from elements of X. Moreover each element $\xi_i \in X$ has m_i links to elements of Q, such that $\sum_i m_i = mx$ (so m is the average). Each element ν_j of Q has q_j in-link and no out-links, so that $\sum_j q_j = qx$. Obviously the two summations amount to the same value so $m = q$.

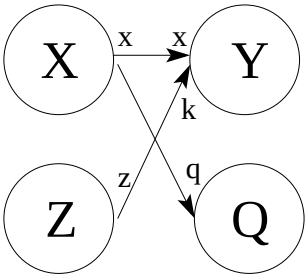


Fig. 6. The Dense XYZQ Model. The figure indicates the 4 node classes and the 3 edge classes. Next to each edge class is indicated the symbol used in the paragraph to denote the corresponding indegree and outdegree of the nodes.

Each element $\eta_j \in Y$ has k_j in-links from elements of Z, such that $\sum_j k_j = kx$ (so k is the average). Each element ζ_i of Z has z_i out-links and no in-links, so that $\sum_i z_i = zx$, and for every i, $z_i \leq |Y| = x$. Obviously the two summations amount to the same value so $z = k$.

Let $u \in X$, thus $N^+(u) = Y \cup Q_u$ where $Q_u \subset Q$ is the set of successors of u in Q and $|Q_u| = m_u$. Let y be an element of $N^+(u)$. If $y \in Y$ then $N^-(y) = X \cup Z_y$ where Z_y is the set of predecessors of y in Z, and $|Z_y| = k_y$. So $d^-(y) = x + k_y$. If $y \in Q_u$ then $N^-(y) = X_y$, where X_y is the set of predecessors of y in X. $d^-(y) = q_y$.

$$B(u) = \sum_{y \in N^+(u)} d^-(y) = \sum_{y \in N^+(u), y \in Y} d^-(y) + \sum_{y \in N^+(u), y \in Q_u} d^-(y) = \sum_{i=1}^{x} (x + k_i) + \sum_{j=1}^{m_u} q_j$$

$$A_1(y) = \sum_{w \in N^-(y), y \in Y} d^+(w) = \sum_{i=1}^{x} (x + m_i) + \sum_{j=1}^{k_y} z_j$$

$$A_2(y) = \sum_{w \in N^-(y), y \in Q_u} d^+(w) = \sum_{i=1}^{q_y} (x + m_i)$$

$$A(u) = \sum_{y \in Y} A_1(y) + \sum_{y \in Q_u} A_2(y) = \sum_{y=1}^{x} [\sum_{i=1}^{x} (x + m_i) + \sum_{j=1}^{k_y} z_j] + \sum_{y=1}^{m_u} \sum_{i=1}^{q_y} (x + m_i)$$

simplifying:

$$B(u) = x^2 + kx + \sum_{j=1}^{m_u} q_j$$

$$A(u) = x^3 + mx^2 + \sum_{y=1}^{x} \sum_{j=1}^{k_y} z_j + \sum_{y=1}^{m_u} \sum_{i=1}^{q_y} (x + m_i)$$

Consider the sum: $\sum_{y=1}^{x} \sum_{j=1}^{k_y} z_j$ this corresponds to taking all elements of Y following the lints to nodes in Z and sum up the out degree of each node in Z as many times as the number of times the node is visited, that corresponds to its out-degree, so we establish:

$$\sum_{y=1}^{x} \sum_{j=1}^{k_y} z_j = \sum_{j=1}^{x} z_j^2$$

Under the assumption that $z \leq x$ and that $z_j \leq x$ for each j we can conclude that this summation is lower bounded by xz^2 and upper bounded by $x^2 z$. Imposing that also $q_j \leq x$ for every j we have that the term $\sum_{j=1}^{m_u} q_j$ is at most $m_u x$, and the term $\sum_{y=1}^{m_u} \sum_{i=1}^{q_y} (x + m_i) \leq m_u(x^2 + mx)$. To summarize we have:

$$x^3 + mx^2 + x^2 k + m_u(x^2 + mx) \geq A(u) \geq x^3 + mx^2 + xk^2,$$

$$x^2 + kx \leq B(u) \leq x^2 + kx + m_u x,$$

and $d^+(u) = x + m_u$. The condition $|Err(u)| \leq \epsilon$ is equivalent to:

$$|B(u)d^+(u) - A(u)| \leq \epsilon B(u)d^+(u).$$

If $B(u)d^+(u) \geq A(u)$ then the left hand side of the inequality is bounded from above by choosing a lower bound for $A(u)$ and an upper bound for $B(u)$ thus after simplification we get:

$$|B(u)d^+(u) - A(u)| \leq x^2(k + m_u - m) + x(m_u + m_u k + m_u^2 - k^2).$$

The leading term multiplying the x^2 factor is equal to the one we have in the sparse case and similar considerations hold. In particular, if $m_u = 0$ the above quantity

is minimized when $k - m = k^2/x$. We still have a cancelation effect, although less strong than before.

If $B(u)d^+(u) \leq A(u)$, we bound from above the quantity $A(u) - B(u)d^+(u)$ by upper bounding $A(u)$ and lower bounding $B(u)$, thus after simplifications we get

$$|A(u) - B(u)d^+(u)| \leq mx^2 + m_u x(m-k).$$

The leading term of the right-hand side of the above inequality,as a polynomial in x, mx^2, is less than the leading term of $\epsilon B(u)d^+(u)$ when $m \leq \epsilon x$. Thus in this second case we need to have a low average noise m to detect the community.

5.3 An asymptotic convergence Theorem for Criterion 2

In particular we will introduce a model that captures the link structure of the pair of sets $\left(\widetilde{\mathrm{N}}(X), \mathrm{N}^+(\widetilde{\mathrm{N}}(X))\right)$ and show some sufficient conditions for which we can prove the asymptotic convergence of Criterion 2.

5.3.1 *The XYZQPRW model.* Consider the pair of sets of nodes $[\widetilde{\mathrm{N}}(X), \mathrm{N}^+(\widetilde{\mathrm{N}}(X))]$. They form a superset for the pair $[\widetilde{\mathrm{N}}(u), \mathrm{N}^+(\widetilde{\mathrm{N}}(u))]$ for all $u \in X$. We will define a decomposition of this pair into seven sets so that the calculation of Criterion 2 for every $u \in X$ is influenced only by the nodes in those sets. We refer the reader to figure 7 for visual help. Let the sets X and Y form the complete bipartite community (fans and centers). Let Z be the set of nodes in $\widetilde{\mathrm{N}}(X) \setminus X$ that have links also to Y, let P be the set of nodes in $\widetilde{\mathrm{N}}(X) \setminus X$ that have no links to Y. The three sets X, Z and P are mutually disjoint and together form a partition of $\widetilde{\mathrm{N}}(X)$: $\widetilde{\mathrm{N}}(X) = X \cup Z \cup P$. Let Q be the set of nodes in $\mathrm{N}^+(X) \setminus Y$. Let R be the set of nodes in $\mathrm{N}^+(\widetilde{\mathrm{N}}(X)) \setminus (Y \cup Q)$ that have some links from Z. Let W be the set of nodes in $\mathrm{N}^+(\widetilde{\mathrm{N}}(X)) \setminus (Y \cup Q)$ that have no links from Z (thus links only from P). The four sets Y, Q and R and W are mutually disjoint and form a partition of $\mathrm{N}^+(\widetilde{\mathrm{N}}(X))$: $\mathrm{N}^+(\widetilde{\mathrm{N}}(X)) = Y \cup Q \cup R \cup W$. The two sets $[\widetilde{\mathrm{N}}(X), \mathrm{N}^+(\widetilde{\mathrm{N}}(X))]$ are not necessarily disjoint. Consider now all possible 12 classes of edges connecting those sets There are 8 possible non-empty classes of links: $X \to Y$, $X \to Q$, $Z \to Y$, $Z \to Q$, $Z \to R$, $P \to Q$, $P \to R$, and $P \to W$. The following link classes are empty: $X \to R$ because such node in R by definition is a node of Q (that is a contradiction), $X \to W$ because such node in W by definition is a node of Q (that is a contradiction), $Z \to W$ because such node in W by definition is a node of R (that is a contradiction), and $P \to Y$ because such node in P by definition is a node of Z (that is a contradiction). See figure 7 for an overall picture.

5.3.2 *The proof of convergence.*

The XYZQPRW decomposition is completely general. In order to derive useful results we need to place restrictions to the cardinality of the sets of nodes and edges involved however we wish to retain sufficient generality. We will assume that the seven sets are roughly comparable in size to $|X| = x$. Next we will assume that, while the class of links $X \to Y$ is of cardinality x^2, all other classes have a number of edges $o(x^2)$, or, equivalently, that the average edge density is $o(x)$ for each other class. Note that so far we impose rather natural conditions, in particular we do not constraint the distribution of the edges and of the node degrees, but only their averages. What we will show is that, even if a few nodes can produce high noise,

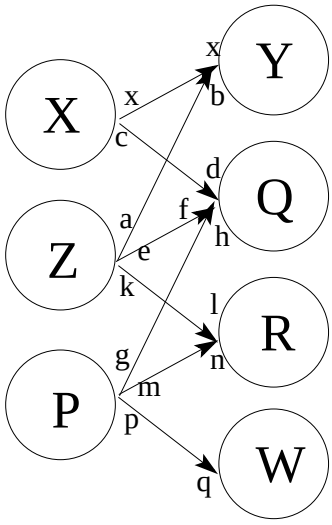


Fig. 7. The XYZQPRW Model. The figure indicates the 7 node classes and the 8 edge classes. Next to each edge class is indicated the symbol used in the proof of Theorem 1 to denote the corresponding indegree and outdegree of the nodes.

when the average noise is low the criterion is asymptotically correct.

THEOREM 1. *In the XYZQPRW model, let* $|X| = x$*, let all other sets be of cardinality* $O(x)$*, let the average edge density for the class* (X, Y) *be equal to* x*, and let the average density be* $o(x)$ *for all other classes, then, as* $x \to \infty$*, there exists* $u \in X$ *such that the* $Err(u)$ *tends to 0.*

PROOF. We will need some notation. For simplicity of calculation, w.l.o.g., let us set all cardinalities to the same value x: $|X| = |Y| = |Z| = |Q| = |P| = |R| = |W| = x$, since at the end these constant factors do not influence the proof. Also note that x is a bound to the indegree and outdegree of each node for any edge class. For the edge class $X \to Y$ the outdegree of every node in X is x, the indegree of every node in Y is x. For the edge class $Z \to Y$, each node i in Z has outdegree a_i each node j in Y has indegree b_j with $\sum_{i=1}^{x} a_i = \sum_{j=1}^{x} b_j = bx$. For the edge class $X \to Q$, each node i in X has outdegree c_i each node j in Q has indegree d_j with $\sum_{i=1}^{x} c_i = \sum_{j=1}^{x} d_j = cx$. For the edge class $Z \to Q$ each node i in Z has outdegree e_i and each node j in Q has indegree f_j with $\sum_{i=1}^{x} e_i = \sum_{j=1}^{x} f_j = ex$. For the edge class $P \to Q$, every node i in P has outdegree g_i and each node j in Q has indegree h_j with $\sum_{i=1}^{x} g_i = \sum_{j=1}^{x} h_j = hx$. For the edge class $Z \to R$ every node i in Z has outdegree k_i and every node j in R has indegree l_j with $\sum_{i=1}^{x} k_i = \sum_{j=1}^{x} l_j = kx$. For the edge class $P \to R$ every node i in P has outdegree m_i and every node j in R has indegree n_j with $\sum_{i=1}^{x} m_i = \sum_{j=1}^{x} n_j = nx$. For the edge class for the family $P \to W$ every i node in P has outdegree p_i and every node j in W has indegree q_j with $\sum_{i=1}^{x} p_i = \sum_{j=1}^{x} q_j = px$.

To summarize: node i in X has outdegree $x+c_i$, node i in Z has outdegree $a_i+e_i+k_i$, node i in P has outdegree $g_i + m_i + p_i$, node j in Y has indegree $x + b_j$, node j in Q has indegree $d_j + f_j + h_j$, node j in R has indegree $l_j + n_j$, and node j in W

has indegree q_j. All quantities $c_i, a_i, e_i, k_i, g_i, m_i, p_i, b_j, d_j, f_j, h_j, l_j, n_j$, and q_j are integers ranging from zero to x.

Let $u = x_i$ be a node in X such that $c_i = o(x)$. Since by hypothesis $c = o(x)$ and c is the mean value of a sum of positive terms, there must exist a value $c_i \leq c = o(x)$. By an abuse of notation we indicate with c_u the corresponding value. Thus $d^+(u) = x + c_u$.

Let $N^+(u) = Y \cup Q_u$ where $Q_u \subset Q$ is the set of successors of u in Q and $|Q_u| = c_u$.

Let y be a generic element of $N^+(u)$. If $y \in Y$ then $N^-(y) = X \cup Z_y$ where Z_y is the set of predecessors of y in Z, and $|Z_y| = b_y$. Therefore $d^-(y) = x + b_y$. If $y \in Q_u$ then $N^-(y) = X_y \cup Z_y, \cup P_y$, where X_y is the set of predecessors of y in X, Z_y is the set of predecessors of y in Z, P_y is the set of predecessors of y in P. Therefore $d^-(y) = d_y + f_y + h_y$.

The function $B(u)$ is

$$B(u) = \sum_{y \in N^+(u)} d^-(y) = \sum_{i=1}^{x}(x+b_i) + \sum_{j=1}^{c_u}(d_j + f_j + h_j) = x^2 + bx + \sum_{j=1}^{c_u}(d_j + f_j + h_j)$$

Clearly it holds that $Q_u \subset Q$, therefore the summation over Q_u is upper bounded by taking all nodes in Q and taking the sums of all indegrees. This value is $dx+fx+hx$, that is $o(x^2)$. So using the hypothesis we obtain that $B(u) = x^2 + o(x^2)$. We split the computation of $A(u)$ into two parts:

$$A(u) = \sum_{y \in Y} A_1(y) + \sum_{y \in Q_u} A_2(y)$$

where

$$A_1(y) = \sum_{w \in N^-(y), y \in Y} d^+(w) = \sum_{i=1}^{x}(x+c_i) + \sum_{i=1}^{b_y}(a_i+e_i+k_i) = x^2 + cx + \sum_{i=1}^{b_y}(a_i+e_i+k_i)$$

and

$$\sum_{y \in Y} A_1(y) = \sum_{y \in Y}\left[x^2 + cx + \sum_{i=1}^{b_y}(a_i + e_i + k_i)\right] = x^3 + cx^2 + \sum_{y \in Y}\sum_{i=1}^{b_y}(a_i + e_i + k_i)$$

The residual summation has this interpretation: follow all links from Y to Z and select each node in Z as many times as its outdegree from Z, with a weight equal to its total outdegree: so it is equal to $\sum_{i=1}^{x} a_i(a_i+e_i+k_i) \leq x^2(a+e+k) = o(x^3)$. Since also cx^2 is $o(x^3)$ we have that the first part of the formula for $A(u)$ is $x^3+o(x^3)$. Compute:

$$A_2(y) = \sum_{w \in N^-(y), y \in Q_u} d^+(w) = \sum_{w \in X_y} d^+(w) + \sum_{w \in Z_y} d^+(w) + \sum_{w \in P_y} d^+(w) =$$

$$= \sum_{w \in X_y} (x + c_w) + \sum_{w \in Z_y} (a_w + e_w + k_w) + \sum_{w \in P_y} (g_w + m_w + p_w)$$

and

$$\sum_{y \in Q_u} A_2(y) \leq \sum_{y \in Q} A_2(y) =$$

$$= \sum_{y \in Q} \sum_{w \in X_y} (x + c_w) + \sum_{y \in Q} \sum_{w \in Z_y} (a_w + e_w + k_w) + \sum_{y \in Q} \sum_{w \in P_y} (g_w + m_w + p_w)$$

The first summation has this interpretation: follow all links from Q to X, so each node of X is visited c_i times, and pick any node with weight $x + c_i$, so it is equal to $\sum_{i=1}^{x} c_i(x + c_i) \leq 2cx^2 = o(x^3)$. The second summation has this interpretation: follow all links from Q to Z picking node i e_i times, with weight equal to the total outdegree of the node, so it is equal to $\sum_{i=1}^{x} e_i(a_i + e_i + k_i) \leq x^2(a + e + k) = o(x^3)$. The third summation has this interpretation: follow all links from Q to P picking node i g_i times, with weight equal to the total outdegree of the node, so it is equal to $\sum_{i=1}^{x} g_i(g_i + m_i + p_i) \leq x^2(g + m + p) = o(x^3)$. Thus the second part of the formula for $A(u)$ is $o(x^3)$, and overall $A(u) = x^3 + o(x^3)$.

Plugging in the computed values in the error function, since $c_u = o(x)$ we obtain that $Err(u) \to 0$, when $x \to \infty$. The proof follows (almost) identically if we assume the size of all sets to be $O(x)$ rather than exactly x. □

6. ANALYZING OVERLAPPING COMMUNITIES

6.1 Analysis of communities with overlapping centers

Here we define the following a simple model for overlapping communities, the (X, Y, V, W, K) model, in order to estimate the effect of this configuration on the function $Err(u)$ for $u \in X$. We assume X and V disjoint, ant that $|X| = x$, $|V| = v$ to be the two fan sets. We assume $Y \cup K$ and $W \cup K$ to be the two sets of centers, with intersection K. $|Y| = y$, $|W| = w$, $|K| = k$. To simplify the calculations we assume full density (i.e. all edges present), no other error sources present and that sets of fans and centers have the same cardinality, therefore we have the relationships:$y + k = x$ and $w + k = v$.

Let $u \in X$, the set $N^+(u) = Y \cup K$. Let $h \in N^+(u)$, If $h \in Y$, then $N^-(h) = X$, thus $d^-(h) = x$. If $h \in K$ then $N^-(h) = X \cup V$, thus $d^-(h) = x + v$.

$$B(u) = \sum_{h \in N^+(u)} d^-(h) = yx + k(x + v)$$

$$A_1(h) = \sum_{w \in N^-(h), h \in Y} d^+(w) = x^2$$

$$A_2(h) = \sum_{w \in N^-(h), h \in K} d^+(w) = x^2 + v^2$$

$$A(u) = \sum_{h \in Y} A_1(h) + \sum_{h \in K} A_2(h) = yx^2 + k(x^2 + v^2)$$

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)} = 1 - \frac{yx^2 + k(x^2+v^2)}{x(yx + k(x+v))} = \frac{kv(x-v)}{x^3 + kvx}$$

We have that $Err(u) \leq \epsilon$ when

$$kv(x-v) \leq \epsilon(x^3 + kvx)$$

that is:

$$k(1 - (u/x) - \epsilon) \leq \epsilon(x/u)x.$$

w.l.o.g. we can assume that $x \geq u$, thus $(x/u) = \alpha > 1$, and $(u/x) = 1/\alpha \leq 1$. Since $h(1 - 1/\alpha - \epsilon) \leq h(1 - 1/\alpha)$, the above inequality is implied by

$$k(1 - 1/\alpha) \leq \epsilon\alpha x.$$

that is:

$$k \leq \epsilon x \left(\frac{\alpha}{1 - 1/\alpha}\right)$$

The function $\frac{\alpha}{1-1/\alpha}$ for $\alpha > 1$ diverges for $\alpha \to 1$ and $\alpha \to \infty$. It is always positive with a unique minimum at $\alpha = 2$, of value 4. Thus, the condition $Err(u) \leq \epsilon$ is implied by $k \leq 4\epsilon x$. Since always $k \leq x$ we have that for $\epsilon = 0.25$ the condition is always satisfied. We conclude that in this "pure" case large overlaps on the set of centers can be handled fairly well by criterion 2.

Consider a scenario in which the community $(X, Y \cup K)$ is a legitimate one, while the community $(V, W \cup K)$ is a spam community set up with the purpose of covering up a link farm. For a value of $\epsilon = 0.25$ we may discover first either community (depending on the order in which the nodes are processed). However, a second pass of the algorithm will detect also the second community.

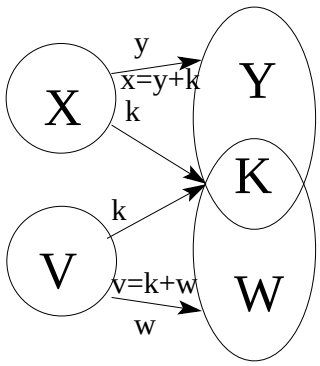


Fig. 8. The XYVWK Model for dense subgraphs sharing centers.

6.2 Analysis of communities with overlapping fans

We have sets (X,Y, V, W, K) with sets of centers Y and W, with $|Y| = y$ and $|W| = w$. The set of fans for Y is the set $X \cup K$ and the set of fans for W is

$V \cup K$, so it holds $x + k = y$ and $v + k = w$. Let $u \in X$, $N^+(u) = Y$, for $h \in Y$, $N^-(h) = X \cup H$, $d^-(h) = y$.

So

$$B(u) = \sum_{h \in N^+(u)} d^-(h) = y^2$$

$$A_1(h) = \sum_{w \in N^-(h), h \in Y} d^+(w) = xy + k(y + w)$$

$$A(u) = \sum_{h \in Y} A_1(h) = y(xy + k(y + w))$$

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)} = 1 - \frac{y(xy + k(y + w))}{y^3} = -\frac{kw}{y^2}$$

Thus $|Err(u)| \leq \epsilon$ when $kw \leq \epsilon y^2$. Thus the test fails when when both the overlap set K is large and the second community has a set of centers V larger than Y. However in this case, by a symmetric argument for a node $u \in V$ it will hold that $|Err(u)| \leq \epsilon$ when $ky \leq \epsilon w^2$. The worst case for the two inequalities derived is when $y = w$ since they both converge to $k \leq \epsilon y$.

Ir remains to consider now the case when $|Y| = |V|$, i.e. $y = w$, and apply the test to a node $u \in K$. For $u \in K$, $N^+(u) = Y \cup W$. If $y \in Y$, $N^-(y) = X \cup H$ and $D^-(y) = x + h = y$. If $y \in W$, $N^-(y) = V \cup H$ and $D^-(y) = v + h = y$.

$$B(u) = \sum_{h \in N^+(u)} d^-(h) = y^2 + w^2$$

$$A_1(h) = \sum_{w \in N^-(h), h \in Y} d^+(w) = xy + k(y + w)$$

$$A_2(h) = \sum_{w \in N^-(h), h \in W} d^+(w) = vw + k(y + w)$$

$$A(u) = \sum_{h \in Y} A_1(h) + \sum_{h \in W} A_2(h) = y[xy + k(y + w)] + w[vw + k(y + w)]$$

$$Err(u) = 1 - \frac{A(u)}{B(u)d^+(u)} = 1 - \frac{y^3 + w^3 + 2kyw}{(y + w)(y^2 + w^2)}$$

Now, using the extra condition $y = w$, we get that $|Err(u)| \leq \epsilon$ when $k \geq y(1-2\epsilon)$.

Note that when $y = w$, for $\epsilon = 1/3$, the two ranges obtained (i.e. $k \leq \epsilon y$ and $k \geq y(1 - 2\epsilon)$) cover all possible values for k, thus we can always detect one of the three dense bipartite communities. Again, multiple passes allows to find them all.

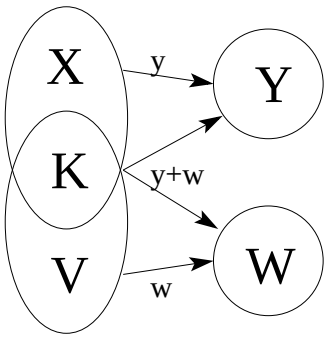


Fig. 9. The XYVWK Model for dense subgraphs sharing fans.

7. TESTING EFFECTIVENESS

By construction algorithms RobustDensityEstimation and ExtractCommunity return a list of dense subgraphs (where size and density are controlled by the parameters t and ϵ). Using standard terminology in Information Retrieval we can say that full precision is guaranteed by default. In this section we estimate the recall properties of the proposed method. This task is complex since we have no efficient alternative method for obtaining a guaranteed ground truth. Therefore we proceed as follows. We add some links in the graph representing the Italian domain of the year 2004, so to create new dense subgraphs. Afterwards, we observe how many of these new "communities" are detected by the algorithm that is run blindly with respect to the artificially embedded community. The number of edges added is of the order of only 50,000 and it is likely that the nature of a graph with 100M edges is not affected.

In the first experiment, about detecting bipartite communities, we introduce 480 dense bipartite subgraphs. More precisely we introduce 10 bipartite subgraphs for each of the 48 categories representing all possible combinations of number of fans, number of centers, and density over a number of fans is chosen in $\{10, 20, 40, 80\}$; number of centers chosen in $\{10, 20, 40, 80\}$; and density randomly chosen in the ranges $[0.25, 0.5]$ (low), $[0.5, 0.75]$ (medium), and $[0.75, 1]$ (high).

Moreover, the fans and centers of every new community are chosen so that they don't intersect any community found in the original graph nor any other new community. The following table (Table I) shows how many added communities are found in average over 53 experiments. For every one of the 48 types, the maximum recall number is 10.

In the second experiment, about detecting quasi-cliques , we introduce ten quasi-cliques for each of 12 classes representing all possible combinations over: number of pages in $\{10, 20, 30, 40\}$, and density randomly chosen in the ranges $[0.25, 0.5]$, $[0.5, 0.75]$, and $[0.75, 1]$. The following table (Table II) shows how many such quasi-cliques are found in average over 70 experiments. Again the maximum recall number per entry is 10.

The cleaned .it 2004 graph used for the test has an average degree roughly 6 (see Section 8). A small bipartite graph of 10-by-10 nodes or a small clique of 10 nodes at 50% density has an average degree of 5. The breakdown of the degree-counting heuristic for these low thresholds is easily explained with the fact that these small

# Centers	Low density				Med. density				High density			
80	0	5.2	9.6	10	1.2	8.4	9.7	10	5.7	8.6	9.5	9.8
40	0	5.4	9.5	9.9	0.7	8	9.7	9.9	5.4	8.6	9.7	9.8
20	0	2.7	5.4	6	0.9	7.9	9.6	9.9	4.6	8.4	9.6	9.9
10	0	0	0	0	0.1	0.8	1.9	3.2	3.3	6.5	9	9.7
	10	**20**	**40**	**80**	**10**	**20**	**40**	**80**	**10**	**20**	**40**	**80**
	# of Fans				# of Fans				# of Fans			

Table I. Number of added bipartite communities found with `threshold=8` depending on number of fans, centers, and density.

# Pages	Low	Med	High
40	9.6	9.8	9.7
30	8.5	9.4	9.3
20	3.6	7.6	8.3
10	0	0.1	3.5
	Low	**Med**	**High**
	Density		

Table II. Number of added quasi-clique communities found with `threshold=8` depending on number of pages and density.

and sparse communities are effectively hard to distinguish from the background graph by simple degree counting.

The analysis in Section 5, and in particular Theorem 1, give some sufficient conditions for the detection of a community using Criterion 2, under certain not too restrictive hypotheses. In applying this theory to the experimental results, one has to keep in mind two caveats. First of all, Criterion 2 is computed for each single node u and involves only sets of nodes "reachable" from u, while the analysis in Section 5 is based on the ensamble of nodes reachable from a set of nodes X. Secondly, a theoretical explanation for the non-detectability of a community by criterion 2 must take into account only nodes reachable form a single node u, and must control all the error sources (i.e. density and noise) simultaneously, and is thus harder to derive. For this reasons in most cases we do not find a comprehensive single-feature discriminant between detected and non-detected communities. However, in certain cases something can be noticed. An examination of the small communities embedded using the XYZQPRW model reveals, for example, that the size of the sets P and W, relative to X and Y, is often a critical parameter. For roughly 80% of the small found communities, $|P| + |W|$ is of the same order of $|X|$ (within a factor 5), while this is true only only for 50% of the small not-found communities. The set Q is usually a small set. The size of the sets Z and R is less critical for these samples: we find small communities even when Z and R are order of magnitude larger than X and Y.

8. LARGE COMMUNITIES IN THE WEB

In this section we apply our algorithm to the task of extracting and classifying real large communities in the web.

8.1 Data set

For our experiments we have used data from The Stanford WebBase project [Cho and Garcia-Molina 2000] and data from the WebGraph project [Boldi and Vigna 2004; Boldi et al. 2004]. Raw data is publicly available at `http://law.dsi.unimi.it/`. More precisely we apply our algorithm on three graphs: the graph that represents a snapshot of the Web of the year 2001 (118M pages and 1G links); the graph that represents a snapshot of the Italian domain of the year 2004 (41M pages and 1.15G links); the graph that represents a snapshot of the United Kingdom domain of the year 2005 (39M pages and 0.9G links).

Since we are searching communities by the study of social links, we first remove all *nepotistic links*, i.e., links between two pages that belong to the same domain (this is a standard cleaning step used also in [Kumar et al. 1999b]). Once these links are removed, we remove also all *isolated* pages, i.e., pages with both outdegree and indegree equal to zero. Observe that we don't remove anything else from the graph, for example we don't need to remove small outdegree pages and large indegree pages, as it is usually done for efficiency reasons, since our algorithm handles these cases efficiently and correctly. We obtain the reduced data sets shows in Table III.

Web 2001	20.1M pages	59.4M links	av deg 3
`.it` 2004	17.3M pages	104.5M links	av deg 6
`.uk` 2005	16.3M pages	183.3M links	av deg 11

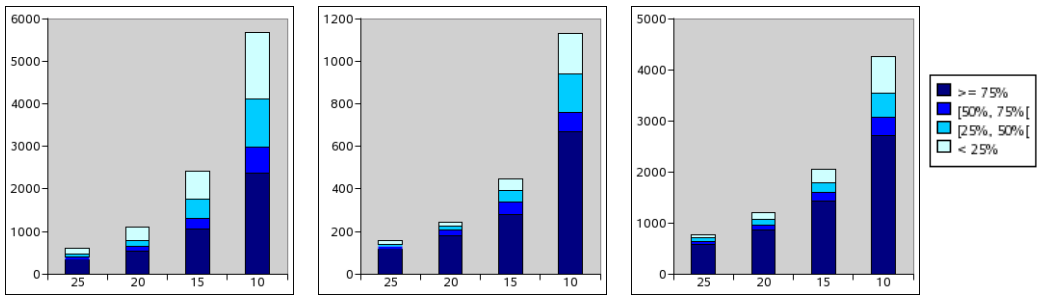
Table III. The reduced data sets. Number of nodes, edges and average degree.

The cleaning phase is completely independent of the two additional parameters of the community finding algorithm (i.e, t and ϵ). In the cleaning phase we do remove nepotistic links (a property of the input web graph, not dependent of any parameter), and we remove nodes of indegree and outdegree degree zero after the removal of nepotistic links (again independent of t). Because of this, the clean graph that serves as input to the community finding algorithm, and also as a model of background noise in the synthetic experiments, is independent of the auxiliary parameters t and ϵ.

8.2 Communities extraction

Figure 10 presents the results obtained with the three graphs presented before. The y axe shows how many communities are found, and the x axe represents the value of the parameter threshold. Moreover communities are partitioned by density into four categories (shown in grey-scale) corresponding to density intervals: [1,0.75],]0.75, 0.5],]0.5, 0.25],]0.25, 0.00].

Table IV reports the time needed for the experiments with an Intel Pentium IV 3.2 Ghz single processor computer using 3.5 GB RAM memory. The data sets, although large, were in a cleverly compressed format and could be stored in main memory. The column "# loops" shows the average number of iterative refinement done for each community in Algorithm ExtractCommunity. Depending on the fan out degree threshold, time ranges from a few minutes to just above two hours for the most intensive computation. In Table V we report, for different values of the threshold t and different data sets, the number of times the filter based on Criterion 2 indicates the possible presence of a community but the subsequent invocation of



(a) Web 2001 (b) Italy 2004 (c) United Kingdom 2005

Fig. 10. Number of communities found by Algorithm RobustDensityEstimation as a function of the degree threshold. The gray scale denotes a partition of the communities by density.

the routine ExtractCommunity fails to find one. In other words, this is the false-positive count of the output of the first filter phase. Also we give the percentage of the false-positive count over the total number of positive indication (each resulting in an invocation of ExtractCommunity). For decreasing values of t the number of false positives increases in absolute terms but decreases in percentage. For example, at value $t = 10$ more than 92% of the times the indication of the filter is correct and a legitimate community is found, thus the computational cost is charged to the output. Only in at most 8% of the times a positive indication from the filter does not result in the discovery of a community, thus in this case the computational cost cannot be charged on the output. Note that this false-positive rate of the first stage does not impact much on the algorithm's efficiency nor on the effectiveness. The false positives of the first stage are caught anyhow by the second stage.

Interestingly in Table VI it is shown the coverage of the communities with respect to the nodes of sufficiently high degree. In two national domains the percentage of nodes covered by a community is above 90% for national domains, and just below 60% for the web graph (of 2001). Table VII shows the distribution of size and density of communities found. The web 2001 data set seems richer in communities with few fans (range [10-25]) and poorer in communities with many fans (range ≥ 100) and this might explain the lower coverage.

Table VII shows how many communities are found with the threshold equal to 10, in the three data sets in function of number of fans, centers, and density. Low, medium and high densities are respectively the ranges $[0.25, 0.5]$, $[0.5, 0.75]$, and $[0.75, 1]$.

9. VISUALIZATION OF COMMUNITIES

The compressed data structure in [Boldi and Vigna 2004] storing the web graph does not hold any information about the textual content of the pages. Therefore, once the list of url's of fans and centers for each community has been created, a non-recursive crawl of the WWW focussed on this list of url's has been performed in order to recover textual data from communities.

Thr.	Web 2001			Italy 2004			Uk 2005		
	# com.	# loops	Time	# com.	# loops	Time	# com.	# loops	Time
10	5686	2.7	2h12m	1099	2.7	30m	4220	2.5	1h10m
15	2412	2.8	1h03m	452	2.8	17m	2024	2.6	38m
20	1103	2.8	31m	248	2.8	10m	1204	2.7	27m
25	616	2.6	19m	153	2.8	7m	767	2.7	20m

Table IV. Some measurements of performance: Number of communities found, average number of cleaning loops per community, and total computing time.

	Web 2001		Italy 2004		Uk 2005	
Thresh.	Num.	perc.	Num.	perc.	Num.	perc.
10	364	6%	34	3%	377	8%
15	135	5%	24	5%	331	14%
20	246	18%	24	9%	526	30%
25	148	19%	4	3%	323	30%

Table V. Absolute number and percentage of calls to ExtractCommunity that do not return a community as output.

Thr.	Web 2001			Italy 2004			Uk 2005		
	# Total	# in Com.	%	# Total	# in Com.	%	# Total	# in Com.	%
10	984 290	581 828	59	3 331 358	3 031 723	91	4 085 309	3 744 159	92
15	550 206	286 629	52	2 225 414	2 009 107	90	3 476 321	3 172 338	91
20	354 971	164 501	46	1 761 160	642 960	37	2 923 794	2 752 726	94
25	244 751	105 500	43	487 866	284 218	58	2 652 204	2 503 226	94

Table VI. Coverage of communities found in the web graphs. The leftmost column shows the threshold value. For each data set, the first column is the number of pages with $d^+ > t$, and the second and third columns are the number and percentage of pages that have been found to be a fan of some community.

What we want is to obtain an approximate description of the community topics. The intuition is that the topic of a community is well described by its centers. As good summary of the content of a center page we extract the text contained in the title tag of the page. We treat fan pages in a different way. The full content of the page is probably not interesting because a fan page can contain different topics, or might even be part of different communities. We extract only the anchor text of the link to a center page because it is a good textual description of the edge from the fan to a center in the community graph. For each community we build a weighted set of words getting all extracted words from centers and fans. The weight of each word takes into account if a word cames from a center and/or a fan and if it is repeated. All the words in a stop word list are removed. We build a flat clustering of the communities. For clustering we use the furthest-point-first (FPF) algorithm described in [Geraci et al. 2007]. As a metric we adopt the Generalized Jaccard

Web 2001 - 5686 communities found at t=10

# Centers	[10, 25[			[25, 50[			[50, 100[			≥ 100		
≥ 100	92	21	49	24	5	8	7	2	8	6	1	11
[50, 100[	185	35	48	38	11	26	9	7	16	11	9	22
[25, 50[	247	54	136	52	28	89	17	6	52	13	14	100
[10, 25[	167	68	437	13	29	217	1	20	163	17	23	347
	low	med	high	low	med	high	low	med	high	low	med	high
	Density			Density			Density			Density		

# of Fans

Italy 2004 - 1099 communities found at t=10

# Centers	[10, 25[			[25, 50[			[50, 100[			≥ 100		
≥ 100	17	3	11	3	1	5	2	2	0	2	1	12
[50, 100[	32	2	14	14	2	4	5	1	2	3	4	15
[25, 50[	28	15	33	10	2	18	5	7	16	19	11	69
[10, 25[	14	5	42	1	3	26	1	2	34	5	11	247
	low	med	high	low	med	high	low	med	high	low	med	high
	Density			Density			Density			Density		

# of Fans

United Kingdom 2005 - 4220 communities found at t=10

# Centers	[10, 25[			[25, 50[			[50, 100[			≥ 100		
≥ 100	24	5	18	17	4	15	10	3	14	11	5	51
[50, 100[	63	23	55	14	21	34	19	11	42	24	22	81
[25, 50[	76	23	151	28	18	159	16	7	68	51	22	273
[10, 25[	43	30	299	7	8	266	8	11	159	34	44	705
	low	med	high	low	med	high	low	med	high	low	med	high
	Density			Density			Density			Density		

# of Fans

Table VII. Distribution of the detected communities depending on number of fans, centers, and density, for $t = 10$.

distance (a weighted form of the standard Jaccard distance).

For each cluster we wish to discover the discriminating words (In future referred as *keywords*). This is a task akin to feature selection in text/web information retrieval. For this purpose for each word in the cluster we sum the score (according to the GJD scores) of all its occurrences in the cluster and select a set of words having highest global score. We refer to this as a the "local keyword selection" since it is done independently for each cluster. We perform also a "global keyword selection" based on maximizing the *information gain* [Cover and Thomas 1991] of the keywords sets. If a keyword appears in two or more cluster the information gain is used to establish for which of them the keyword is more discriminant. For

a term t and cluster c, information gain is defined as:

$$IG(t,c) = \sum_{a\in\{t,\bar{t}\}} \sum_{b\in\{c,\bar{c}\}} P(a,b) \log \frac{P(a,b)}{P(a)P(b)},$$

where the probabilities $P(.,.)$ and $P(.)$ are relative to the random choice of a community. Intuitively, IG measures the amount of information that one variable contains on the other; when t and c are independent, $IG(t,c) = 0$. The IG formula is the sum of four component: two of them represent the "positive correlation" between the variables, while the other represent the "negative correlation" between the variables. In our case, we use IG to select, for each cluster, keywords that are *representative* of the cluster. This means that we are interested only to the positive correlation from the IG formula therefore we drop the factor denoting negative correlation, yielding the modified version:

$$IG_m(c,t) = P(t,c) \log \frac{P(t,c)}{P(t)P(c)} + P(\bar{t},\bar{c}) \log \frac{P(\bar{t},\bar{c})}{P(\bar{t})P(\bar{c})}.$$

This paper focusses on the algorithmic principles and testing of a fast and effective heuristic for detecting large-to-medium size dense subgraphs in the web graph. The examples of clusters reported in this section are to be considered as anecdotical evidence of the capabilities of the Community Watch System.

In Table VIII we show clusters of communities, ranked by cumulative edge count, found by the Community Watch tool in the data-set UK2005 among those communities detected with threshold $t = 25$ (767 communities). Further filtering of communities with too few centers reduces the number of items (communities) to 636. The full listing can be inspected by using the `Community Watch` web interface publicly available at `http://comwatch.iit.cnr.it`. The "Category" labels are assigned manually, after visual inspection of each cluster. This procedure requires a very reasonable human effort (a few man/hours), and, although it entails a degree of subjective judgement, it is sufficient for a static analysis. Some more dynamic applications might benefit from an automation of this classification step via supervised learning techniques and/or the use of computational linguistics techniques (e.g. WordNet `http://wordnet.princeton.edu/`).

The homogeneity score has been assigned as follows. Using the extracted group key-words and a more in depth examination of a few communities in the group we have assigned (manually) a category to the group. Afterwards the communities in the group have been examined manually one-by-one to determine the consistency with the assigned category. The top twenty groups ranked by number of edges are listed in table VIII. Homogeneity is very high or high in 16 out of 20 of the top 20 groups, Medium in 3 out of 20, and Low in just 1 out of 20. Groups in position from 21 to 34 contain groups having a number of edges roughly between 300,000 and 100,000 each, and in general are less homogeneous than the top 20, in particular when they contain a single community that is responsible for most of the edges for that group. Groups ranked from 35 to 62 show a variety of homogeneity scores, including some communities with a tenuous characterization (most likely spam), as well as quite well characterized groups (e.g. adult content).

Cat.	Keyw.	Com	Hom	Cent	Fans	Edges
Telephones	Accessories (0.29) Ericsson (0.17) Phone (0.61)	10	VH	1629	156562	14363477
Cars	Car (0.16) Services (0.26) Shops (1.0)	89	VH	9859	129842	10541928
Antiques	Antiques (0.60) Search (0.07) Selling (0.06)	6	H	1163	13314	2885138
Kitchens	Cookers (0.30) Kitchens (0.61) Sinks (0.30)	13	H	637	48310	2144454
Tourism	Accommodation Bookings (0.09) Hotels (1.0)	37	VH	3047	37259	2003979
Tourism	Cheap (0.25) Holidays (1.0) Villas (0.18)	30	VH	2028	22250	1708089
Tourism	Holiday (0.54) Hotel (1.0) Scotland (0.42)	26	H	1126	34663	1446023
Electronics	Appliance (0.27) Electrical (0.27) Vacuum (0.24)	7	VH	812	25345	1363772
News	Cheshire (0.77) News (1.0) Sport (0.38)	23	H	991	37574	1344061
Telephones	Mobile (0.84) Motorola (0.25) Nokia (0.81)	17	VH	1346	7631	1327239
Finances	Insurance (0.94) Loans (1.0) Mortgages (0.76)	26	M(1)	1040	34774	908505
Shopping	Courses (0.12) Resources (0.07) Sales (0.10)	27	VH(2)	2054	13050	787445
Shopping	Club (0.37) Property (1.0) Services (0.50)	17	M(3)	640	20749	714797
Municipal	Council (0.97) Document (1.0) Services (0.53)	25	H	1492	16227	580206
Computers	Computers (0.90) Hardware (0.33) Software (0.33)	5	H	321	11374	526754
Shopping	Action (0.62) Clothes (0.76) Frames (0.58)	13	L	427	15931	523166
Car/Travel	Car (1.0) Hire (0.27) London (0.16)	34	H	2870	7269	471016
Finances	Credit (0.61) Loans (1.0) Mortgages (0.44)	20	H	2481	6537	453715
Shopping	Category (0.62) Shopping (1.0) Store (0.60)	12	H	519	8833	346646
Shopping	Reviews (0.29) Services (0.36) Updated (0.30)	7	M	292	7229	334350

Table VIII. List of the top 20 groups of communities ranked by total number of edges. For each group of communities we display, a general category, keywords with highest weight for the group, the number of communities in each group. The homogeneity of the communities w.r.t the general category, the cumulative number of centers, fans and edges for each group. Homogeneity is classified in: Very High (VH) when above 90% of the communities are consistent with the category, High (H) when between above 80%, Medium (M) when above 60% and Low (L) when below 60%. Notes: (1) including a community on gambling; (2) generic shopping and services; (3) includes a single very large community selling flowers, gifts, properties,etc.. Data set uk2005 with $t = 25$ of 767 communities. Filtering: centers in [10,..,max], fan degree in [10,..,max], target number of clusters = 80. Resulting in 636 communities organized in 62 groups.

9.1 A Large scale analysis of the uk2005 data set

The data has been analyzed using the following semiautomatic methodology. From the data set uk2005 using threshold 25 and filtering as in Table VIII we have extracted 636 communities. Using `Community Watch` these have been clustered into 62 homogeneous thematic groups labeled by keywords as described above. Manually

Rank	Category	Large comm.	Disperse com.	Total
1	**General Shopping**	13.805.821	3.476.787	17.282.608
2	**Telephony**	14.094.717	529.164	14.623.881
3	**Tourism**	2.775.043	1.245.229	4.020.272
4	**Antiques**	2.502.150	382.988	2.885.138
5	**News**	1.180.855	814.551	1.995.406
6	**Youth Interests**	1.375.605	350.365	1.725.970
7	**Generic Portals**	974.344	-	974.344
8	**Financial Services**	-	957.270	957.270
9	**Training**	492.128	-	492.128
10	**Computers**	490.410	-	490.410
11	**Local Government**	-	340.078	340.078
12	**Adult Contents**	-	89.356	89.356

Table IX. Classification of large and disperse communities by category ranked by number of links. Dataset and filtering as in Table VIII

we have tested the consistency of the single communities with the thematic group, eliminating those communities not well represented by the chosen keywords. Communities that have more than 100.000 edges have been listed as *Large Communities*. Communities smaller than 100.000 edges but included in thematic groups with at least 100.000 edges have been listed as *Disperse Communities*. Thematic groups with less than 100.000 edges have been discarded (except for the Adult Content ones). Afterwards the surviving thematic groups have been associated manually to 12 categories: general Shopping, Telephony, Tourism, Antiques, news, Youth Interests, Generic Portals, Financial Services, Training, Computers, Local Government, Adult Content. These categories were not decided a-priory but emerged from the analysis of the data as the most representative ones. In table IX it is shown the aggregation of communities found in the uk2005 data set into categories and a ranking of the categories by total number of edges.

9.2 State of the art in the classification of web pages and web sites

Collecting together similar web pages based on their textual content (eventually augmented with anchor text) has been done before [Haveliwala et al. 2000] and one of the main objective has been the detection of near duplicates [Broder et al. 1997]. These methods are unsupervised and could detect unifying semantic themes only in some cases. The supervised classification of web sites and web pages has been proposed using several classical classification methods and several features [Lindemann and Littig 2007], [Fang et al. 2006]. Structural properties (e.g. the link structure) is becoming important in such studies [Glover et al. 2002] [Amitay et al. 2003]. An interesting classification mixed methodology is in [Stamou et al. 2006] where the training data is provided implicitly by using hand made directories.

The techniques listed above can handle single sites and small group of pages (in supervised mode) detecting high level functionalities among a set of categories defined at training time. Other techniques could handle large collections of unstructured pages (in unsupervised mode) but the type of inference that could be made were rather week. Our technique falls in between these two extremes. It is unsupervised thus need no initial training, is able to deal with large portions of the web graph and is able to extract high level functional/topical information. Our process of data aggregation produces information that can be validated manually

in the final phase with little effort. We believe that our technique can be valuable for assessing long term macroscopic thematic changes rather than assessing single web site content.

10. CONCLUSIONS AND FUTURE WORK

In this paper we tackle the problem of finding dense sub-graphs of the web-graph. We propose an efficient heuristic method that is shown experimentally to be able to discover about 80% of communities having about 20 fans/centers, even at medium density (above 50%). The effectiveness increases and approaches 100% for larger and denser communities. For communities of less than 20 fans/centers (say 10 fans and 10 centers) our algorithm is still able to detect a sizable fraction of the communities present (about 35%) whenever these are at least 75% dense. Our method is effective for a medium range of community size/density which is not well detected by the current technology. One can cover the whole spectrum of communities by applying first our method to detect large and medium size communities, then, on the residual graph, the Trawling algorithm to find the smaller communities left. The efficiency of the Trawling algorithm is likely to be boosted by its application to a residual graph purified of larger communities that tend to be re-discovered several times.

11. ACKNOWLEDGMENTS

We wish to thanks the anonymous referees for several suggestions and comments leading to a deeper analysis and improvements in the exposition.

REFERENCES

Abello, J., Resende, M. G. C., and Sudarsky, S. 2002. Massive quasi-clique detection. In *Latin American Theoretical Informatics (LATIN)*. 598–612.

Amitay, E., Carmel, D., Darlow, A., Lempel, R., and Soffer, A. 2003. The connectivity sonar: detecting site functionality by structural patterns. In *HYPERTEXT 2003, Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*. 38–47.

Bharat, K., Broder, A. Z., Dean, J., and Henzinger, M. R. 2000. A comparison of techniques to find mirrored hosts on the WWW. *Journal of the American Society of Information Science 51,* 12, 1114–1122.

Bianchini, M., Gori, M., and Scarselli, F. 2005. Inside pagerank. *ACM Trans. Inter. Tech. 5,* 1, 92–128.

Boldi, P., Codenotti, B., Santini, M., and Vigna, S. 2004. Ubicrawler: A scalable fully distributed web crawler. *Software: Practice and Experience 34,* 8, 711–726.

Boldi, P. and Vigna, S. 2004. The webgraph framework I: Compression techniques. In *WWW '04*. 595–601.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. 2000. Graph structure in the web. *Computer Networks 33,* 1-6, 309–320.

Broder, A. Z., Charikar, M., Frieze, A. M., and Mitzenmacher, M. 2000. Min-wise independent permutations. *Journal of Computer and System Sciences 60,* 3, 630–659.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. 1997. Syntactic clustering of the web. In *Selected papers from the sixth international conference on World Wide Web*. Elsevier Science Publishers Ltd., Essex, UK, 1157–1166.

Capocci, A., Servedio, V. D. P., Caldarelli, G., and Colaiori, F. 2004. Communities detection in large networks. In *WAW 2004: Algorithms and Models for the Web-Graph: Third International Workshop*. 181–188.

Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., and Kleinberg, J. 1999. Mining the link structure of the world wide web. *Computer 32,* 8, 60–67.

Cho, J. and Garcia-Molina, H. 2000. WebBase and the stanford interlib project. In *2000 Kyoto International Conference on Digital Libraries: Research and Practice*.

Cover, T. M. and Thomas, J. A. 1991. *Elements of Information Theory*. John Wiley and Sons.

Dourisboure, Y., Geraci, F., and Pellegrini, M. 2007. Extraction and classification of dense communities in the web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*. ACM, New York, NY, USA, 461–470.

Fang, R., Mikroyannidis, A., and Theodoulidis, B. 2006. A voting method for the classification of web pages. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology - Workshops*. IEEE Computer Society, 610–613.

Feige, U. 2002. Relations between average case complexity and approximation complexity. In *Proc. of STOC 2002, Montreal*.

Feige, U. and Langberg, M. 2001. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms 41*, 174–211.

Feige, U., Peleg, D., and Kortsarz, G. 2001. The dense k-subgraph problem. *Algorithmica 29*, 3, 410–421.

Flake, G. W., Lawrence, S., and Giles, C. L. 2000. Efficient identification of web communities. In *KDD '00*. ACM Press, New York, NY, USA, 150–160.

Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. 2002. Self-organization of the web and identification of communities. *IEEE Computer 35*, 3, 66–71.

Geraci, F., Pellegrini, M., Maggini, M., and Sebastiani, F. 2007. Cluster generation and labeling for web snippets:a fast, accurate hierarchical solution. *Internet Mathematics 3*, 4, 413–443.

Gibson, D., Kleinberg, J., and Raghavan, P. 1998. Inferring web communities from link topology. In *HYPERTEXT '98*. ACM Press, New York, NY, USA, 225–234.

Gibson, D., Kumar, R., and Tomkins, A. 2005. Discovering large dense subgraphs in massive graphs. In *VLDB '05*. VLDB Endowment, 721–732.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 7821–7826.

Glover, E. J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D. M., and Flake, G. W. 2002. Using web structure for classifying and describing web pages. In *WWW*. 562–569.

Gulli, A. and Signorini, A. 2005. The indexable web is more than 11.5 billion pages. In *WWW (Special interest tracks and posters)*. 902–903.

Gyöngyi, Z. and Garcia-Molina, H. 2005. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*.

Han, Q., Ye, Y., and et al. 2000. Approximation of dense-k-subgraph. Manuscript.

Hastad, J. 1999. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica 182*, 105–142.

Haveliwala, T. H., Gionis, A., and Indyk, P. 2000. Scalable techniques for clustering the web. In *WebDB (Informal Proceedings)*. 129–134.

Henzinger, M. 2002. Algorithmic challenges in web search engines. *Internet Mathematics 1*, 1, 115–126.

Imafuji, N. and Kitsuregawa, M. 2003. Finding a web community by maximum flow algorithm with hits score based capacity. In *DASFAA 2003*. 101–106.

Ino, H., Kudo, M., and Nakamura, A. 2005. Partitioning of web graphs by community topology. In *WWW '05*. ACM Press, New York, NY, USA, 661–669.

Kautz, H., Selman, B., and Shah, M. 1997. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM 40*, 3, 63–65.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 1999a. Extracting large-scale knowledge bases from the web. In *VLDB '99*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 639–650.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 1999b. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999) 31*, 11–16, 1481–1493.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 2005. Method and system for trawling the world-wide web to identify implicitly-defined communities of web pages. US patent 6886129.

Kumar, S. R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 1999c. Extracting large-scale knowledge bases from the web. In *The VLDB Journal*. 639–650.

Lempel, R. and Moran, S. 2000. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks (Amsterdam, Netherlands: 1999) 33,* 1–6, 387–401.

Lindemann, C. and Littig, L. 2007. Classifying web sites. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*. 1143–1144.

Newman, M. 2003. The structure and function of complex networks. *SIAM Review 45,* 2, 167–256.

Reddy, P. K. and Kitsuregawa, M. 2001. An approach to relate the web communities through bipartite graphs. In *WISE 2001*. 301–310.

Stamou, S., Ntoulas, A., Krikos, V., Kokosis, P., and Christodoulakis, D. 2006. Classifying web data in directory structures. In *Frontiers of WWW Research and Development - APWeb 2006, 8th Asia-Pacific Web Conference*. Lecture Notes in Computer Science, vol. 3841. Springer, 238–249.

Wu, B. and Davison, B. D. 2005. Identifying link farm spam pages. In *WWW '05*. ACM Press, New York, NY, USA, 820–829.