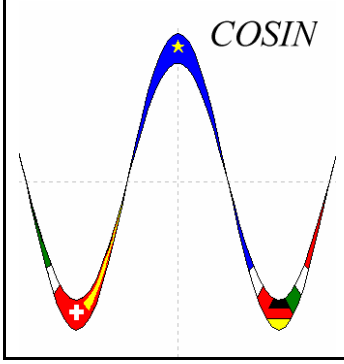

COSIN

IST–2001-33555

Coevolution and Self-Organization in Dynamical Networks

http://www.cosin.org

CyberCommunities in the WWW

Deliverable Number: D23

Delivery Date: September 2005

Classification: Public

Partner Owning: C01(INFM) and CR2(UDRLS)

Contact Authors: Guido Caldarelli *Guido.Caldarelli@roma1.infn.it*

Project Co-ordinator: Guido Caldarelli (INFM) *Guido.Caldarelli@roma1.infn.it*

Partners:

CO1 (INFM) INFM *Italy*

CR2 (UDRLS) Università "La Sapienza" *Italy*,

CR3 (UB) Universitat de Barcelona *Spain*

CR5 (ENS) École Normale Supérieure, Paris *France*

CR7 (UNIKARL) Universität Karlsruhe *Germany*

CR8 (UPSUD) Université de Paris Sud *France*

CR9 (EPFL) Ecole Polytechnique Fédérale de Lausanne *Switzerland*

Project funded by the European Community under the "Information Society Technologies" Programme (1998-2002)

Abstract

Here we present the analysis that has been done by various groups in the project. The first analysis that has been done is based on the state of the art analysis that has been done by broder et al. In their work they analyse some statistical distributions of the whole system. By analysing an even lager crawl we are able to retrieve some of the original results and to propose a description of the system alternative to that of the bowtie. In order to proceed further in the analysis of the communities of the webgraph we present some preliminary results obtained from geographical data sets (obtained outside the project) and thematic datasets collected within the project. Since the complete analysis of both is rather time consuming the results of these studies will be very likely presented in forthcoming projects (namely DELIS). This work is a natural continuation of Deliverable D8.

Contents

Introduction

The Web can be viewed as a directed graph, $G=(V,E)$ where V is the set of (static) HTML pages, and the edges correspond to hyperlinks
We will be using various basic graph theoretic definitions and algorithms that can be found in any graph theory textbook. Here, we only remind the reader of the definitions of strongly and weakly connected components.

A set of nodes S forms a strongly connected component (SCC) in a directed graph, if and only if for every pair of vertices u,v S, there exists a path from u to v, and from v to u. A set of nodes S forms a weakly connected component (WCC) in a directed graph G, if and only if the set S is a connected component of the undirected graph G_u that is obtained by removing the directionality of the edges in G.
Kleinberg et al [1] and Albert and Barabasi [2] demonstrated that the in-degree of the Web graph follows a *power-law* distribution. Later experiments by Broder et al [3] on a crawl of 200M pages from 1999 by Altavista confirmed it as a basic property:
the probability $P(k_u)$ that thethe in-degree k_u of a vertex u is distributed according to a power-law with

$$P[k_u] \quad k_u^{-\gamma} \quad \text{where } \gamma \quad 2.1.$$

The sizes of the SCC components also follow a power-law. The out-degree distribution follows an imperfect power law distribution

Broder et. al [3] studied also the structure of the Web graph, and presented the bow-tie picture. They decomposed the Web graph into the following components (see Figure 1):

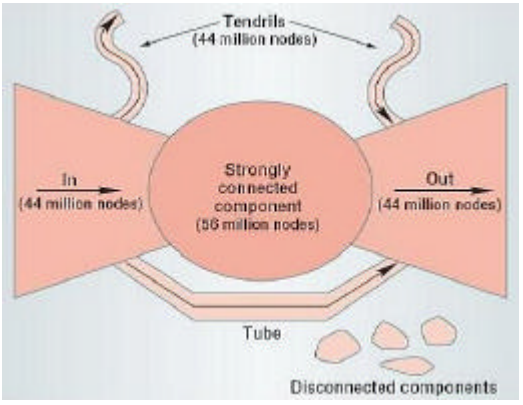

Figure 1 The picture of the Shape of the Web as presented in Ref.[3]

- the CORE, consisting of the largest SCC in the graph;
- the IN, consisting of nodes that can reach the CORE;
- the OUT, consisting of nodes that are reachable from the CORE;
- the TENDRILS, consisting of nodes not in the CORE that are reachable
- from the nodes in IN, or can reach the nodes in OUT;
- the DISC, consisting of the remaining nodes. The fractal structure of the Web has been conjectured in several works.

Dill et al. [4] demonstrated that the Web exhibits self-similarity when considering ``Thematically Unified Clusters" (TUCs), that is, sets of pages that are brought together due to some common trait. They argue that these TUCs form a navigational backbone of the Web.
Thus the Web graph can be viewed as the outcome of a number of similar and independent stochastic processes.

The findings about the structure of the Web generated a flurry of research in the field of random graphs. Given that the standard graph theoretic model of Erd\"os and R\`eny~\cite{erdos} is not sufficient to capture the generation of the Web graph, various stochastic models were proposed~\cite{bar2,pennock,WebAsGraph}. Most of them address the fact that the in-degrees must follow a power-law distribution \cite{bar2}. The \emph{copying model}~\cite{WebAsGraph} generates graphs with multiple bipartite cliques~\cite{KRR}.
%Some of the most popular models
%include the \emph{preferential attachment} model~\cite{bar2}
%and the \emph{copying model}~\cite{krrst}.

Mining the inner structure of the Web

Algorithms

The experiments presented in this section refer to a crawl collected by the WebBase project at Stanford and the dataset is composed by 360 millions of vertices and abut 1.5 billions of edges. Frontier vertices have been pruned to eliminate non significant data. The resulting Graph is then composed by 136 million of nodes and 1.2 billions of edges.
This study has required to node CR2 UDRLS the development of a complete algorithmic methodology for handling very large Web graphs. As a first step authors of Ref. [5] need to identify the individual components of the Web graph. For this they need to be able to perform graph traversals.

The link structure of the Web graph takes several gigabytes of disk space, making it prohibitive to use traditional graph algorithms designed to work in main memory. Therefore, we implemented algorithms that achieve remarkable performance improvements when processing data that are stored on external memory. We implemented *semi-external* algorithms, that use only a small constant amount of memory for each node of the graph, as well as *fully-external* algorithms that use an amount of main memory that is independent of the graph size.

We implemented the following algorithms.

- External versions of Breadth and Depth First search, based on random accesses to the disk, in order to avoid maintaining the data in main memory.
- The traditional traversal algorithms that work in main memory.
- A semi-external graph traversal that allows to determine the verteces reachability for determining vertex reachability using only 2 bits per node. The one bit is set when the node is first visited, and the other when all its neighbors have been visited (we say that the node is ``completed")The algorithm operates on the principle that the order in which the vertices are visited is not important. Starting from an initial set of nodes, it performs multiple passes over the data, each time visiting the neighbors of the non-completed nodes. A semi-external Breadth First Search that computes blocks of reachable nodes and splits them up in layers according to their distance from the root. In a second step, these layers are sorted to produce the standard BFS traversal of the graph
- A semi-external Depth First Search (DFS) that needs 12 bytes plus one bit for each node in the graph. This traversal has been developed following that approach suggested by Sibeyn et al. [6] .
- A semi-external algorithm for computing all SCCs of the graph based on the semi-external DFS.
- An algorithm for computing the largest SCC of the Web graph. The algorithm adopts a heuristic approach that exploits the structural properties of the Web graph to compute the biggest SCC, using a simple reachability algorithm. As a result of the algorithm The algorithm exploits the fact that the largest SCC is a sizable fraction of the Web graph. Thus,

by sampling a few nodes of the graph, we can obtain a node of the largest SCC with high probability.

We can then identify the nodes of the SCC using the reachability algorithm. As an end product we obtain the bow-tie regions of the Web graph, and we are able to compute all the remaining SCCs of the graph efficiently using the semi-external DFS algorithm.

A software library containing a suite of algorithms for generating and processing massive Web graphs is available online **http://www.dis.uniroma1.it/~cosin/**.
A detailed presentation of some of these algorithms and a study of their efficiency has been presented in [7]. A complete description of these algorithms is available in the extended version of this work [8].

Results

As a first step in the analysis of the Web graph, node CR2 UDRLS repeated the experiments of Broder et al. [3] on the macroscopic analysis of the graph computing the in-degree, out-degree and WCC size distributions.

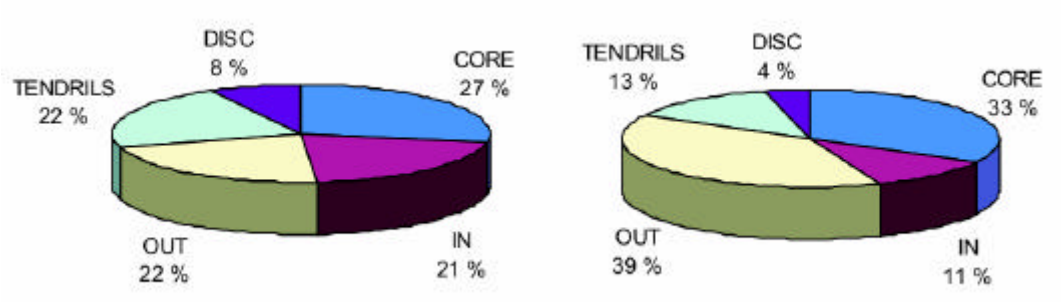

Figure 2: The proportion of the various component of the Web Graph
On the left the AltaVista Graph on the right the analysis on the WebBase Graph

As expected, the in-degrees, and the sizes of SCCs follow a power-law distribution, while the out-degree distribution follows an imperfect power-law.

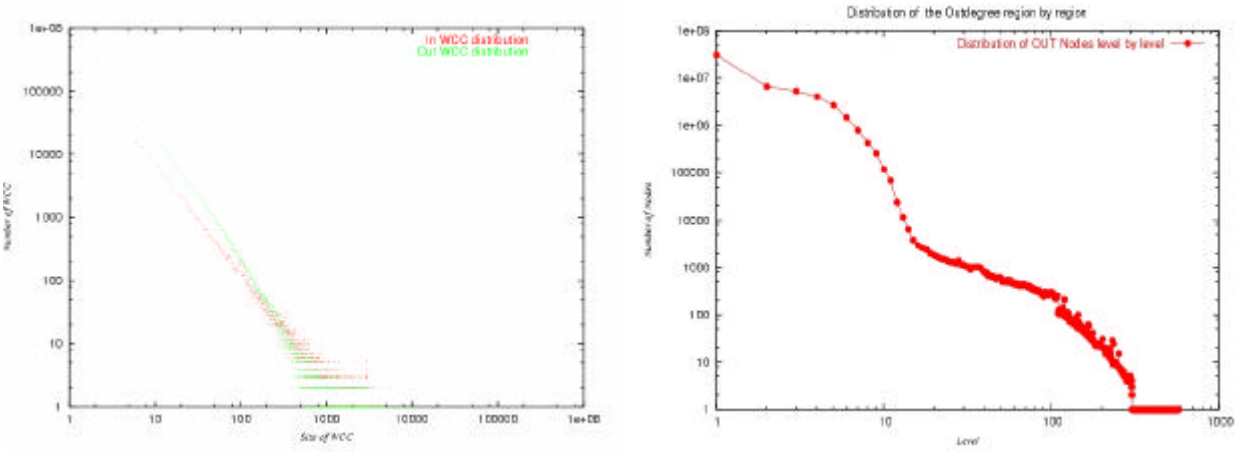

Figure 3 the plot of the Frequency distributions of WCC sizes and out-degree

A complete description of the Work done is given in Ref.[5] Here we only summarize the following facts:

- The inner structure as regards degree distributions and sizes of WCC and SCC is the same for the IN and OUT component.

- The CORE has *entry points* that is to say nodes connected with at least one node in the IN region and has also *exit points* that point at least to a node in the OUT region. *Bridges* are told those vertices that are both entry and exit points. Surprising the vast majority of vertices (~72 %) are EXIT points. This means that about 80% of the core is connected to the external regions, while only 20% belong to inner core region.
- One can also define connectors those vertices of the core that have a single in-coming and out-coming link. A connector forms a petal if if they coincide. Surprisingly only 6% of the CORE vertices are connectors (47% of which are petals).

Putting together the various feature it seems that a different macroscopic picture of the WWW must be done. Instead of Bow-tie a Daisy seem to represent better the shape of the graph,

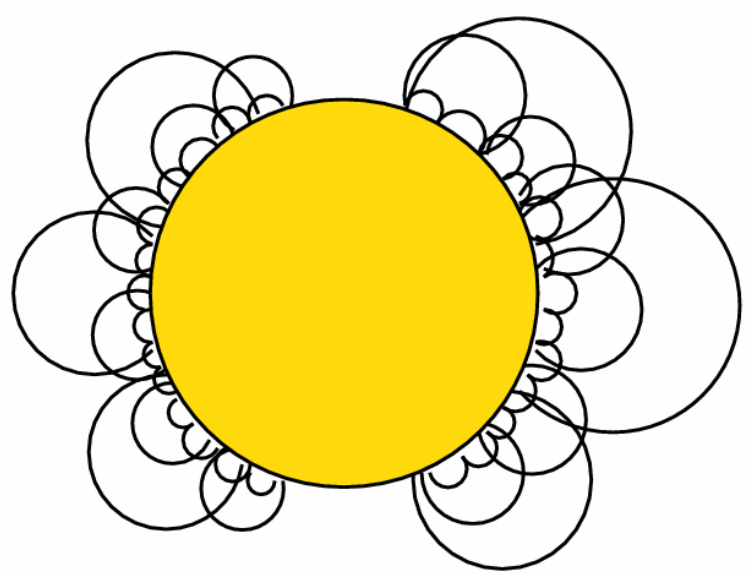

Figure 4 The daisy structure of the WWW

Correlation between vertices the paradox of WWW

A distinctive characteristic of a network is whether its vertices tend to connect to similar or unlike peers, the so-called mixing property. Similarity of vertices is established by comparing some scalar quantity measuring a given qualità of the vertex. Borrowing terms from sociology, networks where properties of neighboring nodes are positively correlated are called *assortative*, while those showing negative correlations are called *disassortative*. Thus, assortative and disassortative mixing patterns indicate a generic tendency to connect respectively to similar or dissimilar pears. A scalar quantity naturally associated to each node in a network is its degree, measuring the number of neighboring nodes. The mixing by degree (MbD) is often measured by looking at how the average degree $<k_{nn}>$ of the nearest neighbors of a node depends on the degree k of the node itself, and is a signature of correlations between other networks quantities. The mixing is assortative when $<k_{nn}>$ grows with k and disassortative when it decreases.
The relevance of MbD lies in that, beyond discriminating among different network morphologies it reflects important structural properties.
Assortative networks are found to be more resilient against the removal of vertexes than disassortative ones. This implies, for example, that, when trying to block infection or opinion spreading within a social network or to protect a computer network against cyber-attacks, different strategies are needed depending on the MbD properties of the underlying network. Moreover, it has recently been observed that the sign of degree correlations affect other properties of complex networks such as synchronization.

Recent studies show that social networks exhibit assortative MbD, whereas technological and biological ones display disassortative MbD[9]. The Word Wide Web (WWW), a paradigmatic example of world-wide collaborative effort among millions of users and publishers, represents an anomaly:
one would expect it to show assortative mixing, similarly to other social and collaborative networks, while it shows evidences of anticorrelations [10], and disassortative MbD [11], which would rather put it in the realm of technological networks.

The aim of the publication realised by node C01 INFM [12] is to demonstrate that in the WWW case, for example, links with different direction have different roles and meanings: the outgoing links are drawn by
individual web--masters, while they have no control on incoming links. A page gains authority from incoming links, while it increases a peer's authority by pointing to it. Nevertheless, the WWW has been often analyzed and modeled as an undirected network for what concerns its mixing properties.

The main result is that, in most cases, assortativity patterns are reversed when the direction of links in a network is taken into account: positive correlations among the degree of a node and the average degree of both upstream and downstream neighbors, considered separately, can disappear or even reverse when the different nature of neighboring sites is ignored and their degree are averaged together. Though this result may appear counterintuitive, the fact that pooling together data of different nature can generate spurious correlations is well known in the statistical literature, and often encountered in social sciences, medical statistics and finance, where, although it contains no logical contradiction, it is known as *Simpson's paradox* [13].

In Ref. [12] Capocci et al. demonstrate the crucial role of link directions in the analysis of mixing patterns in complex networks. Their main result is that assortativity patterns are often reversed once a network is considered as directed. In the growing complex network models we have analyzed, we find positive correlations between the degree of a node and the average degrees of both upstream and downstream nodes, while fictitious correlations emerge when the different nature of the nodes is not taken into account. This is an example of the Simpson's paradox that may occur any time data from different sources are pooled together. The correlation that appears in the pooled data is spurious: a positive correlation between two quantities before pooling results negative after pooling and vice versa. In the particular case of growing networks the degrees of upstream and downstream neighbors of a node are positive correlated with the degree of the node itself, however the correlation with upstream neighbors is much weaker. For increasing degrees, the fraction of weakly correlated neighbors increases. The overall neighbors' average degree can then decrease as a result of the varied proportion, misleadingly suggesting the presence of negative correlations. In the case of BA networks, this effect exactly balances that of positive correlations. Our findings suggest the need for more detailed analysis of real directed networks, such as the WWW, with a special focus on the direction of links between nodes. The counterintuitive properties described above may explain the anomalous exclusion of the WWW from the realm of social networks based on its observed disassortative mixing.

Detecting Communities

Several empirical methods to detect communities have already been proposed. The most successful is an algorithm, introduced by Ref [14] (NG--algorithm), based on the edge betweenness, which measures the fraction of all shortest paths passing on a given link, which is related to the probability that a random walk on the network runs over that link. By removing links with high betweenness, one progressively splits the whole network into disconnected components, until the network is decomposed in communities consisting of single nodes. The outcome of the algorithm is represented by a dendrogram, i.e. a tree--like diagram where each branching corresponds to a splitting event.

Though this method has been shown to be very powerful in cases where some a priori knowledge of the a community structure is given, it has two main disadvantages: firstly, it does not give an indication of the resolution of the clustering, and thus it needs extra information as input (like the expected number of clusters); secondly, its outcome is independent on how sharp the partitioning of the graph is.
An alternative way to tackle the problem, which is the one made by COSIN participants, is by spectral analysis [15]. Previous approaches to graph partitioning from spectral analysis have been mostly developed in the computer science community to the purpose of finding the best allocation of processes on processors in parallel computers, and are based on iterative bisection. These algorithms have the same limitation of the one based on the edge betweenness, since they give no indication of when the bisection should terminate, and thus need extra information on the expected number of communities. Moreover, while repeated bisection is an efficient method for process allocation, when applied to the search of community structure it is by its own nature not guaranteed to reach the best or ``most natural" partition.

The activity of the consortium in this area has been focussed in the interpretation of the spectral analysis of adjacency matrix in terms of minimisation principles.
We first review the spectral properties of some matrices defining the network, and show how some relevant information on the cluster structure is contained in the first few nontrivial eigenvectors. To make it clear how the relation between communities and eigenvectors components arises, we then recast the problem in terms of constrained optimization of some function. The algorithm that we propose, uses correlations among components of different eigenvectors. We formulate the algorithm for undirected weighted networks, and then we propose a modified version suitable to take into account link orientation. Our method allows to correctly detect communities in sharply partitioned graphs. However, in most real cases, one deals with large complex networks, where there is no well defined partition, and the question of finding the community structure can be ill--posed. In such cases, our method can be used to analyze the structure of the network around a given node, or to find a set of nodes well connected to a given one.

As an example of this second application, we test the algorithm on a large scale data--set from a psychological experiment of free word association, where it proves to be successful both in clustering words, and in uncovering mental association patterns.

Detecting community structure by spectral analysis

Spectral methods are based on the analysis of the adjacency matrix $\boldsymbol{A}$, whose element a_{ij} is equal to 1 if i points to j and 0 otherwise. Actually, it is more convenient to study not the adjacency matrix itself but rather either one of two matrices that can be derived from it, namely:

- the Laplacian matrix $\mathbf{L = K - A}$
- the normal matrix $\mathbf{N = K^{-1}A}$.

where $\mathbf{K}$ is the diagonal matrix with elements k_{ii}= degree of node i is the number of nodes in the network. In most approaches, referring to undirected networks, $\mathbf{A}$ is assumed to be symmetric.

The matrix $\mathbf{N}$ has always the largest eigenvalue equal to one, associated to a trivial constant eigenvector, due to row normalization. In a network with an apparent cluster structure, $\mathbf{N}$ has also a certain number *m-1* of eigenvalues close to one, where *m* is the number of well defined communities, the remaining eigenvalues lying a gap away from one. The eigenvectors associated to these first *m-1* nontrivial eigenvalues, also have a characteristic structure: the components corresponding to nodes within the same cluster have very similar values x_i, so that, as long as the partition is sufficiently sharp, the profile of each eigenvector, sorted by components, is step--like. The number of steps in the profile corresponds again to the number m of communities. A similar information is encoded in the non-negative definite Laplacian matrix, where the eigenvalues close to zero are associated to clusters.

The spectrum of the Laplacian matrix has a number of null eigenvalues equal to the number of connected components of the graph. The eigenvectors spanning the kernel of L can be chosen such that they have constant components on one connected subgraph and are zero elsewhere. In a similar way, the normal matrix, whose eigenvalues lie in the range between -1 and 1, has as many unity eigenvalues as the number of connected components of the graph.

Let us think for simplicity to a case with just two connected components. The matrix L will be block diagonal, and the null eigenvalue will be twofold degenerate. As soon as a small perturbation is added to **L** such to couple the two components (in terms of the network this means adding one or more links connecting the two subnets), the degeneracy is removed: there is just one null eigenvalue with trivial constant eigenvector, and the other null eigenvalue is shifted slightly above zero, still a gap away from the remaining eigenvalues. The idea is that this eigenvector, being a small perturbation to one that is constant on each connected component, it is still approximately constant on each component, as long as the perturbation is small enough.

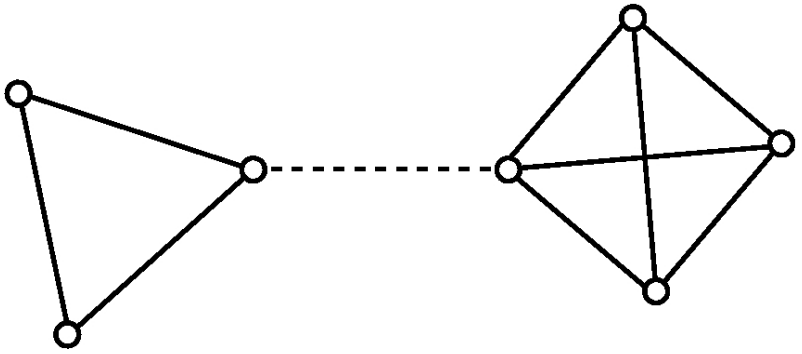

Figure5 Undirected network employed as an example, with two disjoint complete graphs of order 3 and 4.

The graph in Figure 2 is made up of two disjoint completely connected components of order 3 and 4 respectively. In this case, the Laplacian matrix is block diagonal (one 3x3 block and one 4x4 block), with an appropriate labeling of the nodes. The null eigenvalue is then present with twofold degeneraci and the corresponding eigenspace may be spanned by two orthogonal
eigenvectors of the type:

- (1 1 1 0 0 0 0)
- (0 0 0 1 1 1 1)

Next we add one link between two vertices lying in two disjoint components as shown in Figure by the dashed edge. This perturbation removes the degeneracy of the null eigenvalue (in fact we have only one connected component in the graph now), while at the zeroth order the previous two eigenvectors change into a linear combination of them. The result is that now we have two eigenvectors given respectively by:

- (1 1 1 1 1 1 1)
- (a a a -b -b -b -b)

with a and b chosen such to preserve orthogonality with the trivial eigenvector. By observing at the structure of the first non trivial eigenvector we are thus able to discern the community structure of the graph, by simply considering as belonging to the same community the vertices corresponding to the index of this eigenvector with similar values. The situation remains substantially unaltered if more than 2 clear communities are present. In case there are *m* well defined communities in the graph, we would have as zeroth approximation a linear combination of *m* vectors.
We would also have now *m* eigenvalues close to zero, indicating that there are actually *m* clear communities in the graph.
The reasoning with the normal matrix is substantially the same except that the null eigenvalue of the laplacian matrix is to be replaced by the unity and that the eigenvectors need not to be orthogonal as the normal matrix is not symmetric.

Translation into an optimization problem

It is actually possibile to recast the eigenproblem into an optimization problem. With the most general applications in mind, instead of the adjacency matrix **A**, we focus on the weighted matrix **W**, whose elements w_{ij} are assigned the intensity of the link (i,j). We consider undirected graphs first, and then we pass to the most general directed case.

Consider the following constrained optimization problem:
Let $z(\mathbf{x})$ be defined as

$$z(\mathbf{x}) = \frac{1}{2} \sum_{i,j}^{1,\dots,S} (x_i - x_j)^2 w_{ij}$$

where x_i are values assigned to the nodes, with some constraint on the vector **x**, expressed by

$$\sum_{i,j}^{1,\dots,S} (x_i \, x_j \, m_{ij}) = 1$$

where m_{ij} are elements of a given symmetric matrix **M**. The stationary points of z over all **x** subject to this constraint are the solutions of

$$(\mathbf{D} - \mathbf{W})\,\mathbf{x} = \mu\, \mathbf{M}\, \mathbf{x}$$

where D is the diagonal matrix giving the degree of the weighted adjacency matrix and μ is a Lagrange multiplier.

Different choices of the constraint **M** leads to different eigenvalues problems.

- choosing **M=D** leads the eigenvalues problem $\mathbf{D}^{-1}\mathbf{W}\,\mathbf{x} = (1-2\mu)\,\mathbf{x}$, (Normal)
- while **M**=1 leads to $(\mathbf{D}-\mathbf{W})\mathbf{x} = \mu\mathbf{x}$.(Laplacian)

Thus, solving the eigenproblem is equivalent to minimizing the function z with the constraint given, where the x_i's are eigenvectors components. The absolute minimum corresponds to the trivial eigenvector, which is constant. The other stationary points correspond to eigenvectors where components associated to well connected nodes assume similar values.

The study of the eigenvectors profiles and the eigenvalues has practical use only when a clear partition exists, which is rarely the case. In most common occurrences, the number of nodes is too large and the separation between the different communities is rather smooth. Thus communities cannot be simply detected by looking at the first nontrivial eigenvector: the typical eigenvector profile is not step-like, rather it is a smooth curve. We resolve this issue by combining information from the first few eigenvectors, and extracting the community structure from correlations between the same components in different eigenvectors. In this way, one can still efficiently
detect sets of well connected nodes.

When the values of the components of the first non—trivial eigenvector are not well distinct, as they are in the toy example above, the information coming from other eigenvectors can be used to give extra information. The idea is that nodes belonging to the same community, and thus having similar values for the corresponding second eigenvector components, are expected to have distinct values, but again close to each other, in the next few eigenvectors. Thus one has to look at correlations between corresponding components of different eigenvectors, rather than the value of the components itself.
A natural way to identify communities in an automatic manner, is by measuring the correlation

$$r_{ij} = (\langle x_i \, x_j \rangle - \langle x_i \rangle \langle x_j \rangle) / [(\langle x_i^2 \rangle - \langle x_j^2 \rangle)(\langle x_i \rangle^2 - \langle x_j \rangle^2)]^{1/2}$$

where the average <…> is over the first few nontrivial eigenvectors. The quantity r_{ij} measures the community closeness between node i and j. Though the performance may be improved by averaging over more and more eigenvectors, with increased computational effort,

one finds that indeed a small number of eigenvectors suffices to identify the community to which nodes belong, even in large networks.

To detect the community structure in a directed network, one replaces, in the previous analysis, the matrix W with a matrix $\mathbf{Y}=\mathbf{W}\mathbf{W}^{\mathbf{T}}$. This corresponds to replacing the directed network with an undirected weighted network, where nodes pointing to common neighbors are connected by a link, whose intensity is proportional to the total sum of the weights of the links pointing from the two original nodes to the common neighbors.
Then, one performs the analysis on the undirected network as described previously. Thus, the function to minimize in this case is

$$y(\mathbf{x}) = \sum_{i,j,l}^{1,\ldots,S} (x_i - x_j)^2 \, w_{il} \, w_{jl}$$

Defining $\mathbf{Q}$ as the diagonal matrix whose entries are the gneralisation of the degree

$$q_{ij} = \delta_{ij} \sum_{i,j,l}^{1,\ldots,S} w_{il} \, w_{jl}$$

the eigenvalue problem for the analogous of the generalized normal matrix,

$$\mathbf{Q}^{-1} \, Y \, \mathbf{x} = l \, \mathbf{x}$$

is equivalent to minimizing the function y under the constraint

$$\sum_{i,j,l}^{1,\ldots,S} x_i \, x_j \, q_{ij} = 1$$

Tested on simple examples of directed networks, the algorithm associated to the minimization of y, outperforms the one based on the minimization of z.

An example from psychology: the free word association network

To test this spectral correlation-based community detection method on a real complex network, we apply the algorithm to data from a psychological experiment reported in reference [16].
Volunteers participating to the research had to respond quickly by freely associating a word (response) to another word given as input (stimulus), extracted by a fixed subset. The words given as responses were then used iteratively as stimuli. Scientists conducting the research have recorded all the stimuli and the associated responses, along with the occurrence of each association. One can build a network where words are nodes, and directed links are drawn from each stimulus to the corresponding responses, assuming that a link is oriented from the stimulus to the response. Weights are associated to each link, according to the frequencies of each association. The resulting network includes 10616 nodes, with an average in-degree equal to about 7. From this network we construct the weighted adjacency matrix $\mathbf{W}$. In this example, passing to the matrix Y means that we expect stimuli giving rise to the same response to be correlated.

science	1	literature	1	piano	1
scientific	0.994	dictionary	0.994	cello	0.993
chemistry	0.990	editorial	0.990	fiddle	0.992
physics	0.988	synopsis	0.988	viola	0.990
concentrate	0.973	words	0.987	banjo	0.988
thinking	0.973	grammar	0.986	saxophone	0.985
test	0.973	adjective	0.983	director	0.984
lab	0.969	chapter	0.982	violin	0.983
brain	0.965	prose	0.979	clarinet	0.983
equation	0.963	topic	0.976	oboe	0.983
examine	0.962	English	0.975	theater	0.982

Figure 6: The words most correlated to *science*, *literature* and *piano* in the eigenvectors of $\mathbf{Q}^{-1} \, \mathbf{W} \, \mathbf{W}^{T}$. Values indicate the correlation.

This new method detects communities of highly connected nodes within a network, by combining information from the first few eigenvectors, and extracting the community structure from correlations between corresponding components in different eigenvectors. The algorithm applies to the most general case of weighted and directed networks. Unlike previous spectral approaches, this method is not based on iterative bisection. The method correctly detects communities in sharply partitioned graphs, without a priori knowledge of the expected number of communities. Also, in cases where there is no well defined partition, it can still be used to analyze the structure of the network around a given node, or to find a set of nodes well connected to a given one. As an example we have analyzed the weighted directed network obtained from a large data--set from a psychological experiment of free word association. The algorithm proves to be successful in clustering nodes (in this case, words) according to reasonable criteria, and provides an automatic way to extract the most connected sets of nodes to a given one in a set of over 10^4 vertices.

Geographic subsets

The result over the large crawl of WebBase have been tested over 3 differents geographic subsets. experiment with four different crawls. The first three crawls are samples from the Italian Web (the .it domain), the Indochina Web (the .vn, .kh, .la, .mm, and .th domains), and the UK Web (the .uk domain) collected by the "Language Observatory Project" and the "Istituto di Informatica e Telematica" using UbiCrawler [17]. The sizes of the crawls are shown in Table 1.

	Italy	**Indochina**	**Uk**	**Web Base**
Nodes	41.3M	7.4M	18.5M	135.7M
Edges	1.15 G	194.1 M	298.1M	1.18G
Core	29.8 M (72.3%)	3.8 M (51.4%)	1.2M (65.3%)	44.7M (32.9%)
In	13.8 K (0.03%)	48.5K (0.66%)	312.6K (1.7%)	14.4M (10.6%)
Out	11.4M (27.6%)	3.4M (45.9%)	5.9M (31.8%)	53.3M (39.3%)
Tendrils	6.4K (0.01%)	50.4K (0.66%)	139.4K (0.8%)	17.1M (12.6%)
Disc	1.25 K (0%)	101.1K(1.4%)	80.2K (0.4%)	6.2M (4.6%)

TABLE 1. Sizes and bow-tie components for the different crawls and the Alta Vista graph

Essentially the kind of the analysis is the same already done for the fourth largest crawl and that brought to the definition of the daisy model.
Given the fact that the in-degree, out-degree, and SCC size distributions in the IN and OUT components are the same as for the whole Web graph, one could wonder if the Web has a *self-similar* structure [6, 12], that is if the bow-tie structure repeats itself inside the IN and OUT components.
The first indication that the self-similarity conjecture is not true in this case comes from the fact that there exists no sizable SCC in the IN and OUT components that could play the role of the CORE in a potential bow-tie. Moreover we surprisingly discovered that there is no giant weakly connected component\ (WCC) in either of the two components. In fact, there is a large number ofWCCs per component and their sizes follow a power law distribution.
In order to better understand how the nodes in IN and OUT are arranged with respect to the CORE, we performed the following experiment. We condensed the CORE in a single node and we performed a forward and a backward BFS. This allows us to split the nodes in the IN and OUT components in *levels* depending on their distance from the CORE.
In all graphs, the depths of the components are relatively small. Furthermore, most nodes are concentrated close to the CORE. Typically, about 80-90% of the nodes in the OUT component are found within the first 5 layers. For the WebBase graph, although the OUT is much deeper, with 580 levels, more than 58% of its nodes are at distance 1 from the CORE, and 93% are within distance 5. Furthermore, after level 305 there exists only a single chain of nodes that extends until level 580, making the effective depth of the OUT 305.

Therefore, one can conclude that the IN and OUT components are shallow and highly fragmented. They are comprised of several sparse weakly connected components of low depth. Most of their volume consists of nodes that are directly linked to the CORE.
As regards the study of the CORE there are two main aspects:
(1) its relation with the IN and OUT components
(2) its connectivity properties
the first question can be approached by measuring the *entry points* to the CORE (nodes that are pointed to by at least one node in the IN component), the *exit points* (nodes that point to at least one node in the OUT component and the *bridges* (nodes that are both entry and exit points).
Regarding the connectivity, there are few nodes with just one in and out link that could make the CORE weakly connected. These are the *connectors* (or *petals* if the source of the incoming link, and the target of the out-going link are the same node) previously defined. Moreover the CORE seems resilient to targeted attacks performed by deleting not only nodes with total degree bigger than a prefixed threshold but also the k nodes with the highest in-degree and k nodes with the highest out-degree. In the first case, we observe that the threshold on the total degree must become as low as 100 in order to obtain an SCC of size less than 50% of the CORE.
There are two ways to interpret these results. The first is that there are no obvious *failure points* in the CORE, that is, strong hubs or authorities that pull the rest of the nodes together, and whose removal from the graph causes the immediate collapse of the network.
In order to disconnect the CORE you need to remove nodes with sufficiently low degree. On the other hand, note that one can manage to reduce the largest SCC to 35-40%of the original by removing about 1Mnodes. However this is less than 1% of the total nodes. In that sense the CORE is vulnerable to targeted attacks.

Thematic subsets
In order to collect several sets of thematic web pages, node C01 INFM togheter with Centro Studi e Ricerche “E. Fermi” deployed a two steps strategy.

- In the first part of our crawling, it has been used a software to query Google and get 10 thousand web pages in different languages containing a specific word.

- Later a second software has been used in order to follow this procedure:
    1. check if the page contained the specific word chosen, and proceed only if it did
    2. count the outdegree of this page (the outdegree is the number of the links contained in a page)
    3. follow all the links of this page
    4. for each destination page check if it also contained the specific word we chose, and if so count the outdegree

Therefore each set of data is made by a list of web URLs downloaded from Google (they actually contain a specific word with the outdegree of these pages). For each of these URLs are then collected the list of the web pages they pointed to, only if those ones contain the same word. This procedure is repeated for several steps (depth of research). The outdegree of all these pages is recorded and analysed.
While this analysis is only at the beginning we already found some interesting results. By varying the depth of the search, the degree-distribution is almost in any case a power-law. This would suggest that the system is scale-invariant over several orders of magnitude since similar distribution holds also for very local subsets of the system.

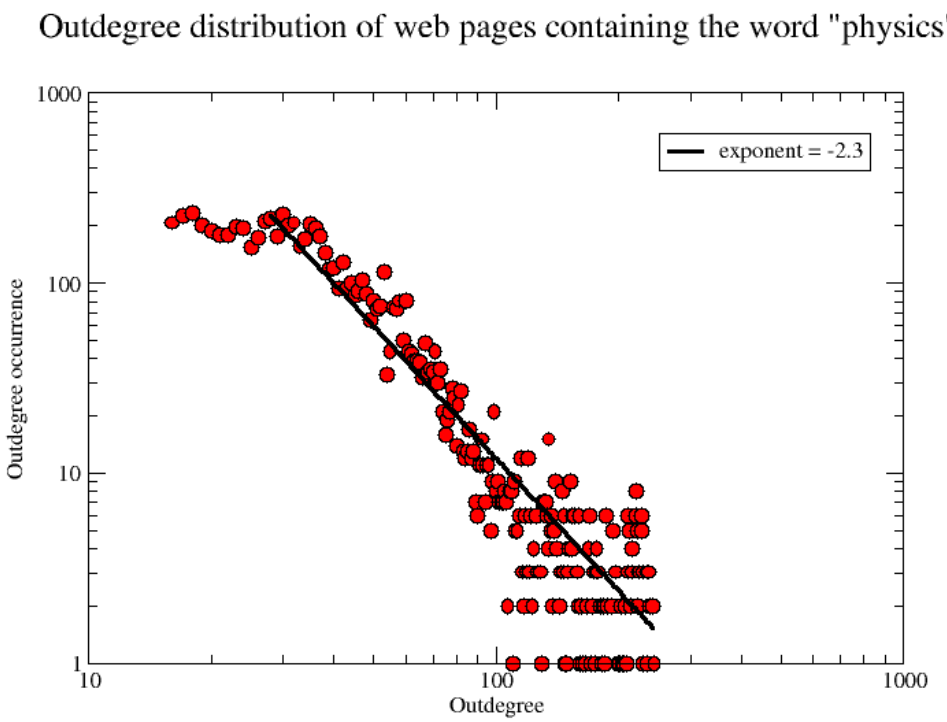

Figure 7 The distribution of the subset of pages (90.000) containing the word physics

As very preliminary results it has also been noticed that the clustering of the pages collected in this way, varies considerably with the topic chosen, thereby calling for a social analysis of the contents.

Bibliography

[1] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, The Web as a graph: measurements, models and methods, *Proc. Intl.Conf. on Combinatorics and Computing*, **1627** 1-18 (1999).

[2] R. Albert, and A.-L. Barabasi, Statistical mechanics of complex networks, *Reviews of Modern Physics*, **74** 47-97 (2002).

[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, S. Stata, A. Tomkins, J. Wiener, Graph structure in the web, *Computer Networks*, **33** 309-320 (2000).

[4] S. Dill, R. Kumar, K. McCurley, S. Rajagopalan, D. Sivakumar, A. Tomkins, Self-similarity in the web, *Proceedings of the 27th VLDB Conference*, 69-78, (2001).

[5] D. Donato, S. Leonardi, S. Millozzi, P. Tsaparas, Mining the Inner structure of the Web *WWW2005*.

[6] J.F. Sibeyn, J. Abello U. Meyer, Heuristic for semi-external depth first search on directed graph. *Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, (2002).

[7] L. Laura, S. Leonardi, S. Millozzi, U. Meyer, Algorithms and Experiments for the Webgraph. *European symposium on algorithms- Lecture Notes in Computer Science*, **2461** (2002).

[8] See deliverable D13 of this project.

[9] M. E. J. Newman *Physical Review E* **67**, 026126 (2003).

[10] M.E.J. Newman , *Physical Review Letters* **89** 208701 (2002).

[11] A. Capocci, G. Caldarelli, P. De Los Rios, *Physical Review E* **68** 047101 (2003).

[12] A. Capocci, F. Colaiori, *ArXiv:cond-mat*0506509 (2005).

[13] E. H. Simpson *Journal of the Royal Statistical Society B* **13** 238 (1951).

[14] M. Girvan and M. E. J. Newman, *Proc. Natl. Acad. Sci*. USA **99**, 8271 (2002).

[15] A. Capocci, V.D.P. Servedio, G. Caldarelli and F. Colaiori, *Physica A* **350** 491-499 (2005).

[16] M. Steyvers, J. B. Tenenbaum, preprint cond-mat/0110012, submitted for publication.

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