Graph Structure of the Web

Module 4

Bowtie Structure of the Web

- Represented the Web as a directed graph where the webpages are treated as vertices and hyperlinks are the edges and included the properties in this graph: diameter, degree distribution, connected components and macroscopic structure.
- The dark-web (part of the Web that is composed of webpages that are not directly accessible (even by Web browsers)) were disregarded.

Power Law

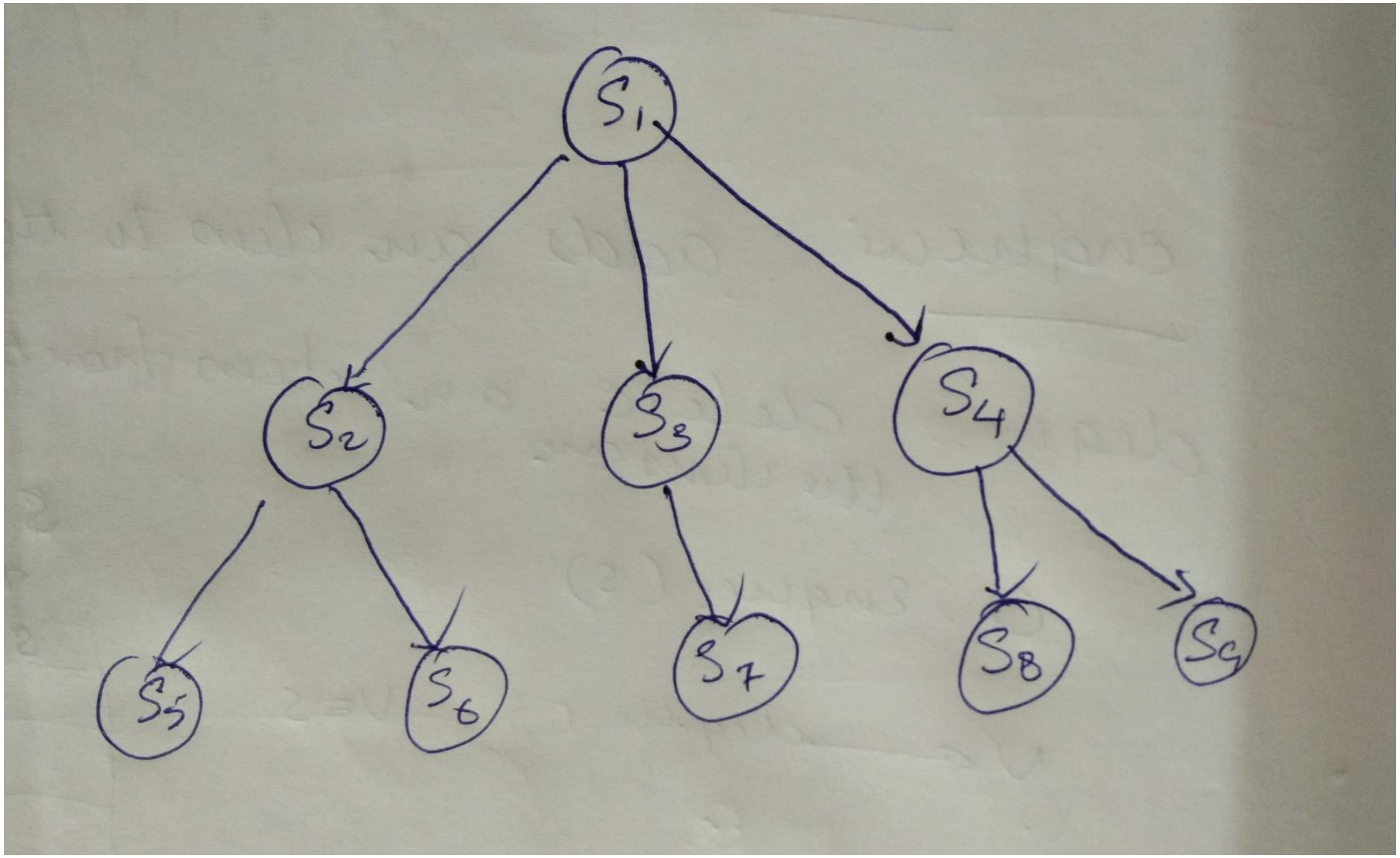
- The power law is a functional relationship between two quantities where a relative change in one quantity results in a proportional relative change in the other quantity, independent of the initial size of those quantities, i.e, one quantity varies as the power of the other.
- Hence the name power law.
- Power law distributions as defined on positive integers is the probability of the value i being proportional to 1/ i_k for a small positive integer k.

Breadth First Search (BFS) Algorithm

- The BFS algorithm takes as input a graph *G(V, E)* and a source vertex *s*.
- The algorithm returns the set of vertices that the source vertex has a path to.
- The Web crawl proceeds in BFS, subject to various rules designed to avoid overloading, infinite paths, spam, time-outs etc.
- Each build is based on crawl data after further filtering and processing.
- Due to multiple starting points, it is possible for the resulting graph to have several connected components

Algorithm 1 BFS algorithm

```
procedure BFS(G(V, E),s)
    Let reachable be an array
    Let Q be a queue
    Q.enqueue(s)
    reachable.add(s)
    while Q is not empty do
        v ← Q.dequeue()
        for each neighbour w of v in V do
            if w is not in reachable then
                Q.enqueue(w)
                reachable.add(w)
            end if
        end for
    end while
    return reachable
end procedure
```



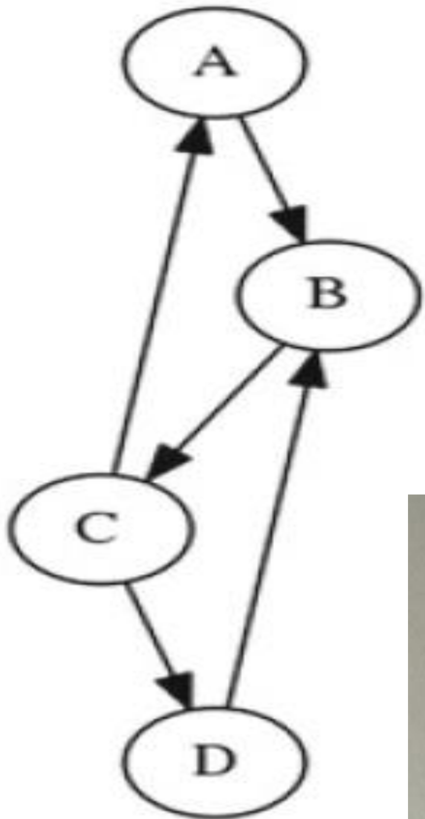
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
S_9

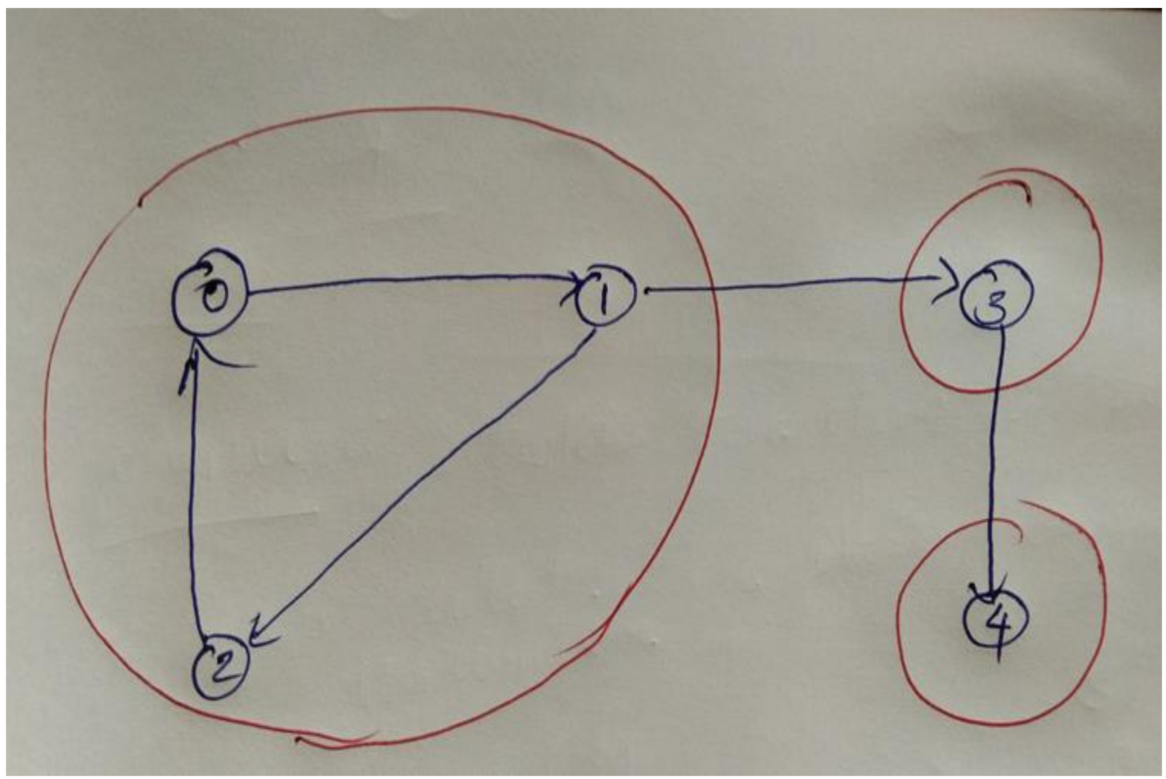
In and Out of a Vertex

- For a graph *G(V, E)*, we define the *In* and *Out* of a vertex $v \in V$ as given
- In(v) = {w ∈ V | there exists a path f rom w to v}
- Out (v) = {w ∈ V | there exists a path f rom v to w}

Strongly Connected Component

- A strongly connected component (SCC) is a set of vertices S such that it satisfies the following conditions:
  - Every pair of vertices in S has a path to one another.
  - There is no larger set containing S that satisfies this property
  - $In(v) = Out\ (v)\ \forall\ v \in V$


A
B
C
D


0
1
2
3
4

Strongly Connected Components (SCC) Algorithm

- The SCC algorithm takes a graph *G(V, E)* as input and returns a list of all the SCCs in this graph as output.
- The SCC of G is computed using the equation
- $In(v) = Out\ (v)\ \forall\ v \in V$
- The function unique returns a list of all the distinct elements in the input list.
  - For instance, unique([1,2,1,3,2,2]) will return [1,2,3].

Algorithm 2 In algorithm

```
procedure In(G(V, E),s)
    Let In be an array
    for each vertex v in V do
        if s in BFS(G(V, E), v) then
            In.add(v)
        end if
    end for
    return In
end procedure
```

Algorithm 3 Out algorithm

```
procedure Out(G(V, E),s)
    Let Out be an array
    Out ← BFS(G(V, E), s)
    return Out
end procedure
```

Algorithm 4 SCC algorithm

```
procedure SCC(G(V, E))
    Let SCC be an array
    for each vertex v in V do
        SCC.add(In(G(V, E), v) ∩ Out(G(V, E), v))
    end for
    return unique(SCC)
end procedure
```

Weakly Connected Components (WCC) Algorithm

- The WCC algorithm computes the list of all WCCs given a graph *G(V, E)* as input.

Algorithm 5 WCC algorithm

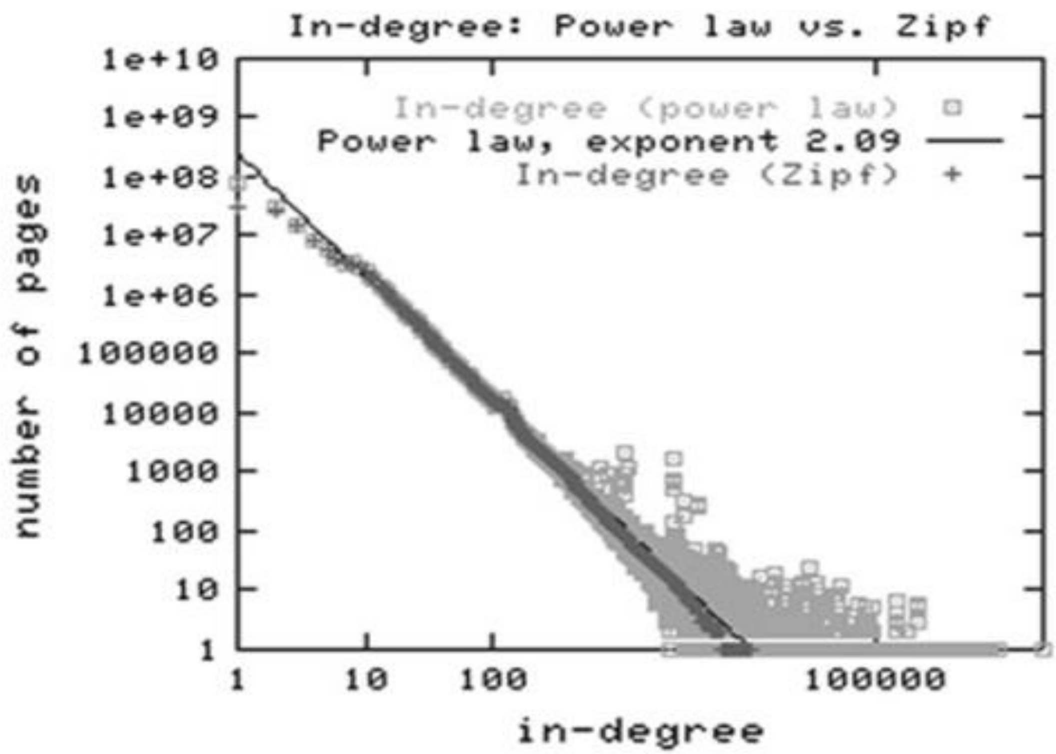
procedure WCC($G(V, E)$)
    Let WCC be an array
    Let $G'(V', E')$ be $G(V, E)$ as an undirected graph
    for each vertex v in V' **do**
        $WCC.add(BFS(G'(V', E'), v)$
    end for
    return $unique(WCC)$
end procedure

Degree Distribution

- The degree distribution of a graph G(V, E), denoted by P(k), is defined as the probability that a random chosen vertex has degree k.
- If $|V|_k$ denotes the number of vertices with degree k,
- $$P(k) = |V|_k \,/\, |V|$$
- The degree distribution is plotted either as a histogram of k vs P(k) or as a scree plot of k vs $|V|_k$

Zipf's Law

- Zipf's law states that the frequency of occurrence of a certain value is inversely proportional to its rank in the frequency table.



In-degree distributions plotted as Power Law and Zipf's Law

- From the giant undirected component, the *DI SCONNECT ED COMPONENT S* and the *SCC*.
- The 100 million vertices whose forward BFS traversals exploded correspond to either the *SCC* component or a component called *IN*.
- Since, *SCC* corresponds to 56 million vertices, this leaves with 44 million vertices (≈ 22%) for IN.
- The 100 million vertices whose backward BFS traversals exploded correspond to either SCC or a component called OUT.
- The remaining vertices which are not accounted were called TENDRILS.

- These components altogether form the bowtie structure of the Web.
- The intuition behind these components is that IN corresponds to those webpages that can reach the SCC, but cannot be reached back from the IN component.
- This possibly contains new webpages that have not yet been discovered and linked to.
- The webpages that are accessible from the SCC, but do have a link back to this

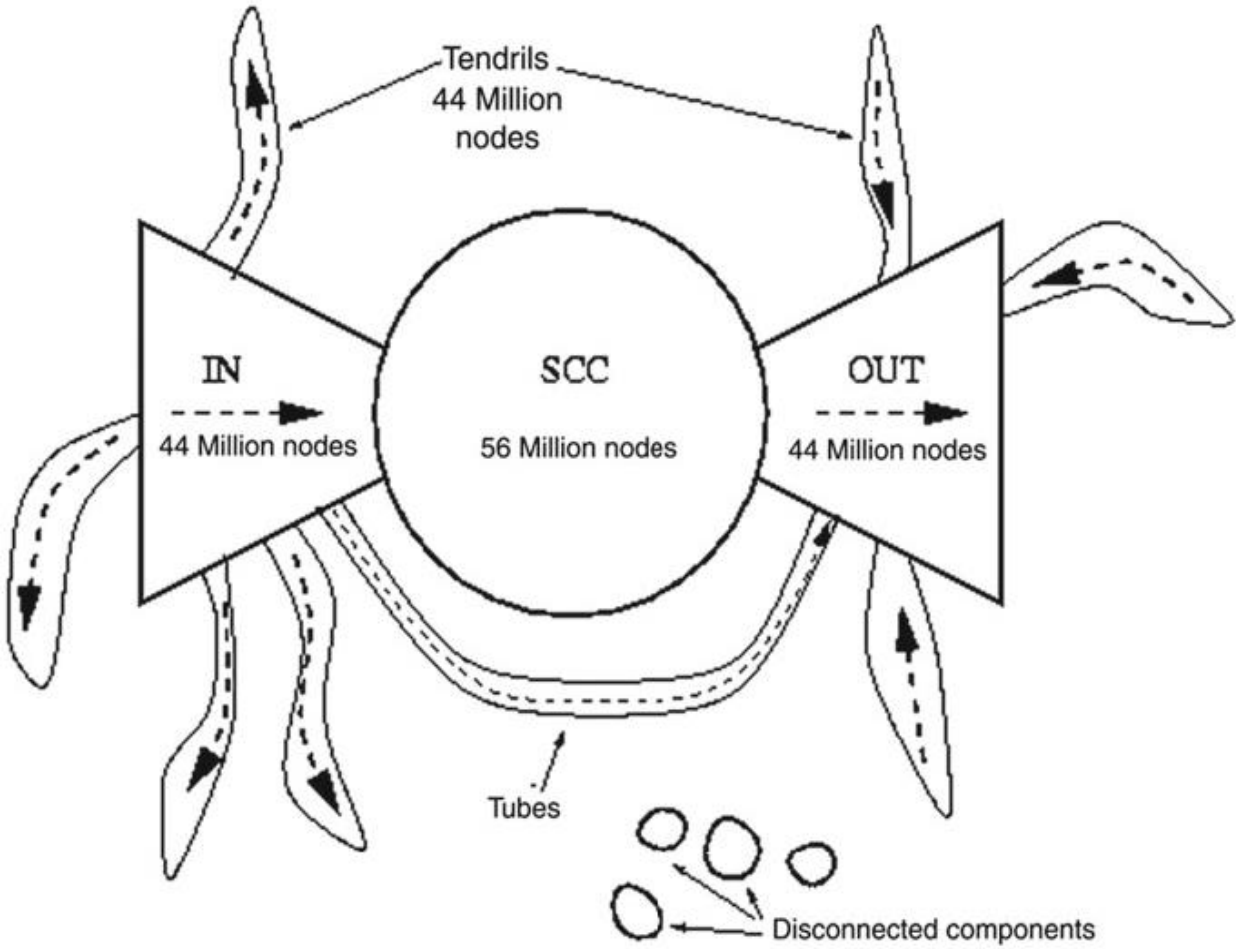


Fig. 2.11 Bowtie structure of the graph of the Web

The Power Laws

- Power-laws are expressions of the form $y\ \alpha\ x^{a}$, where a is a constant, x and y are the measures of interest, and α stands for “proportional to".
- The exponents of the power-laws can be used to characterize graphs.

Rank Exponent R

- From the log-log plots of the out-degree d_v as a function of the rank r_v in the sequence of decreasing out-degree
  - The out-degree, d_v, of a node v is proportional to the rank of this node, r_v, to the power of a constant, R .

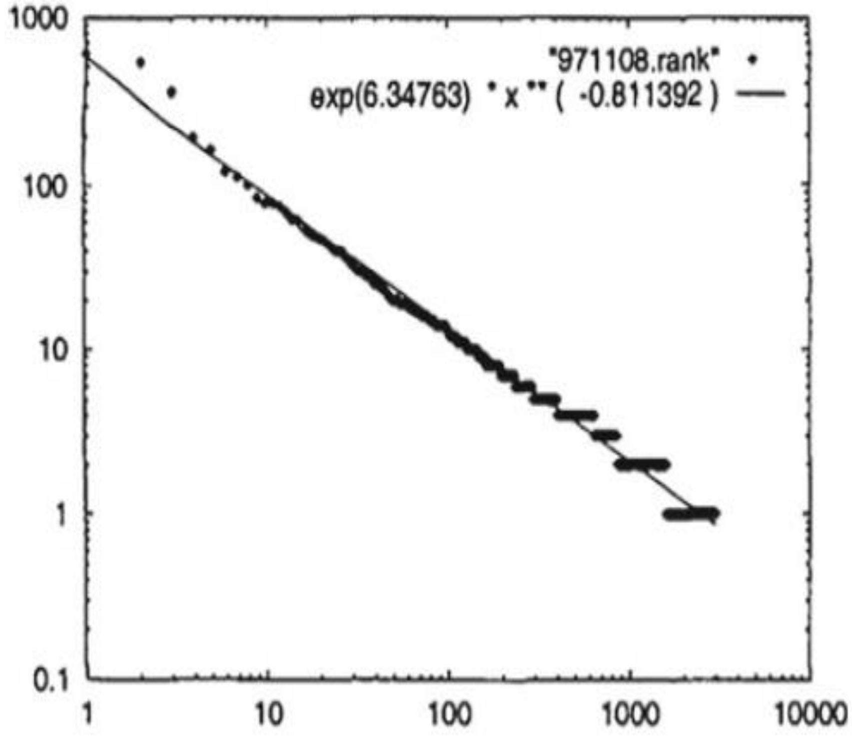
$$d_v \propto r_v^{\mathcal{R}}$$

- If the nodes of the graph are sorted in decreasing order of out-degree, then the rank exponent, R is defined to be *the slope of the node versus the rank of the nodes* in the log-log scale.
- The out-degree, d_v, of a node v is a function of the rank of this node, r_v, and the rank exponent, R,

$$d_v = \frac{1}{N^{\mathcal{R}}} r_v^{R}$$

- The number of edges, |E|, of a graph can be estimated as a function of the number of nodes, |V|, and the rank exponent, R,

$$|E| = \frac{1}{2(\mathcal{R}+1)}\left(1 - \frac{1}{N^{\mathcal{R}+1}}\right)^{|V|}$$



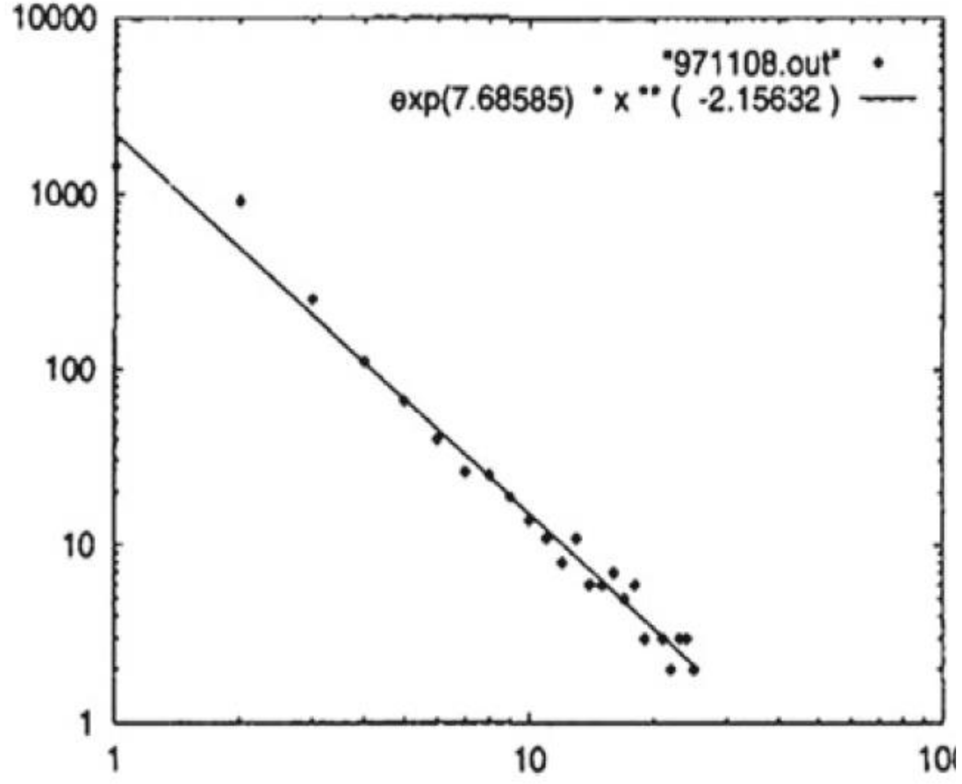
Log-log plot of the out-degree d_v as a function of the rank r_v in the sequence of decreasing out-degree for Int-11-97

Out-Degree Exponent O

- The log-log plots of the frequency f_d as a function of the out-degree d
  - The frequency, f_d , of an out-degree, d, is proportional to the out-degree to the power of a constant, O.

$$f_d \propto d^{O}$$

  - The out-degree exponent, O, is defined to be *the slope of the plot of the frequency of the out-degrees versus the out-degrees in log-log scale*.



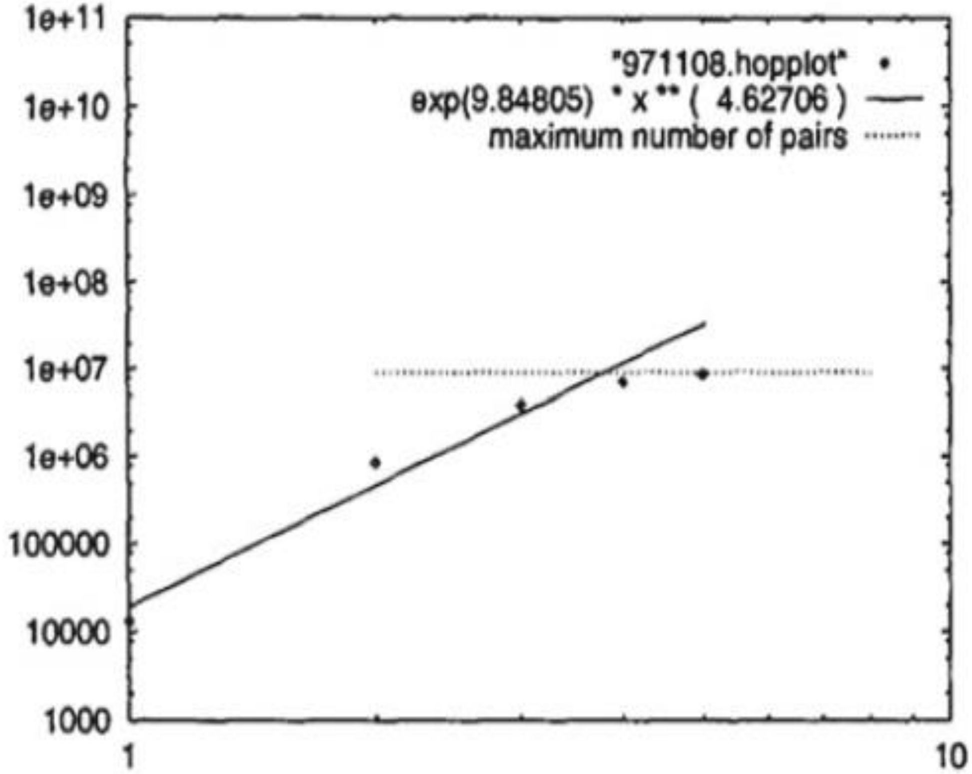
Log-log plot of frequency f_d versus the out degree for Int-11-97

Hop Plot Exponent H

- The total number of pairs of nodes P(h) within h hops, defined as the *total number of pairs of nodes within less or equal to h hops, including self-pairs, and counting all other pairs twice*, is plotted as the function of the number of hops h in log-log scale
  - The total number of pairs of nodes, P(h), within h hops, is proportional to the number of hops to the power of a constant, H

$$P(h) \propto h^{\mathcal{H}}, h << \delta$$

  - where δ is the diameter of the graph.



Log-log plot of the number of pairs of nodes *P(h)* within *h* hopes versus number of hops *h* for Int-11-97

- If we plot the number of pairs of nodes, P(h), within h hops as a function of the number of hops in log-log scale. For h < δ, the slope of this plot is defined to be the hop-plot exponent H
- The number of pairs within h hops is as given in

$$P(h) = \begin{cases} ch^{\mathcal{H}}, h << \delta \\ |V|^2, h \geq \delta \end{cases}$$

- where c = | V | + 2 | E | to satisfy internal conditions.

- Given a graph with |V| nodes, |E| edges and H hop-plot exponent, the effective diameter, δ_{ef} is defined as

- $$\delta_{ef} = \left(\frac{|V|^2}{|V| + 2|E|}\right)^{1/\mathcal{H}}$$

- The average size of the neighborhood, *NN(h)*, within h hops as a function of the hop-plot exponent, H, for h << δ, is as

$$NN(h) = \frac{c}{|V|}h^{\mathcal{H}} - 1$$

- where c = |V| + 2|E| to satisfy internal conditions

- The average out-degree estimate , NN’ (h), within h hops with average out-degree d’, is

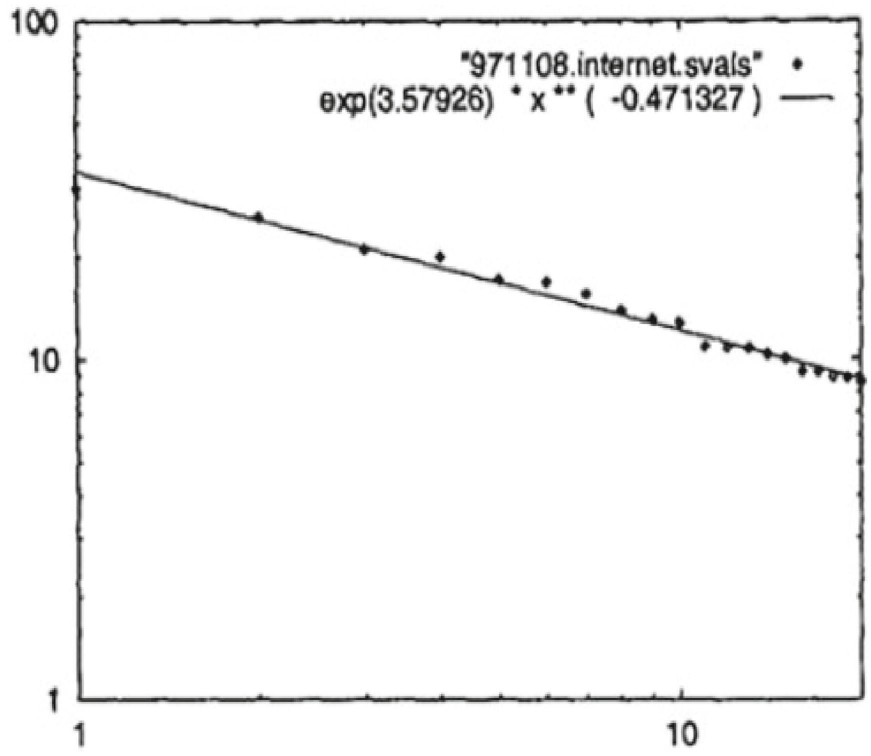
$$NN'(h) = \overline{d}(\overline{d} - 1)^{h-1}$$

Eigen Exponent ε

- The plot of the eigenvalue λi as a function of i in the log-log scale for the eigenvalues in the decreasing order
  - The eigenvalues, λi , of a graph are proportional to the order, i , to the power of a constant, ε

$$\lambda_i \propto i^{\mathcal{E}}$$

  - The Eigen exponent ε is defined as the *slope of the plot of the sorted eigenvalues as a function of their order in the log-log scale.*



Log-log plot of the eigenvalues in decreasing order for Int-11-97