On the Structural Algorithm of Filtering Graphs for Layout

Xiaodi Huang [1,2] Wei Lai[2]

[1]Department of Mathematics and Computing, the University of Southern Queensland, Australia
[2]School of Information Technology, Swinburne University of Technology, Australia
Email: {xhuang, wlai}@it.swin.edu.au

Abstract

When the amount of information in visualization becomes large enough, users can not perceive all elements at the same time. This problem can be solved by removing parts of the information through the process of *Filtering*. In this paper, we present a novel method for filtering a graph by measuring the important role property of a node. The basic idea of this approach is to quantify the importance of a node as the degree to which it has direct and indirect relationships with the other nodes in a graph. All the nodes are ranked according to their Node Importance Scores, and those less important nodes and their associated edges are then removed or invisible. In comparison with the rule_based approach, our approach is a structure_based one that makes use of the linkage of nodes rather than their semantics. It can therefore be applied to filter any kinds of connected graphs. The examples and applications provided have demonstrated that our approach can effectively reduce the visual complexities .

Keywords: Graph Visualization, Filtering, Web graph

1 Introduction

While there are other ways to tackle the layout of large graphs such as clustering, navigation, and zooming, the proposed approach is significantly different to these. Navigation and zooming cannot replace filtering since filtering reduces the size of a focus subgraph and supports logical identification of such a subgraph. Filtering is thus one alternative way to efficiently reduce the visual complexity.

The reasons why a graph should be filtered are:

- There exist "noise" nodes in the graph--defined as those nodes that interfere with the representation of a relation.

Usually, a large graph is automatically generated from an information source. This may unavoidably lead to the creation of "noise" information. For example, the use of a web crawler program easily extracts some unwanted image files, together with html web pages, from a web site when constructing a Web graph.

- Overview first, zoom and filter, then details-on-demand

According to the Visual Information Seeking Mantra [Shneiderman 1996], filtering is an efficient method of handling information on a large scale in that it initially presents an overview of the data and then gradually reveals details of the data.

Huang et al. [Huang, et al. 1998] applied several rules to filter a real Web graph into a simplified tree for a simplified layout. The rules are: *Graph structure based rule*, *Web context structure based rule*, *Information based rule*, *Document structure based rule*, and *Link number based rule*. The use of these rules ensures that the converted graph is a tree, and simplifies the original Web graph by removing some particular nodes and edges. It is obvious that these rules are concerned with both the structure and content of a Web graph.

Simple filtering techniques based on attributes are also reported in Henry's thesis [Henry 1992].

The above approaches have some limitations. Huang et al.'s approach, for instance, is confined to be a specific type of graph, i.e. Web graphs, and does not suffice to handle other types of graphs. Therefore, it is necessary to develop a more sophisticated technique for filtering general graphs.

The approaches to filtering graphs can be roughly classified into *Structure_* and *Content*_oriented. The *Structure_* based approach utilizes the linkages of a graph while the *Content_* based one uses the semantic contents of what the nodes represent. The proposed approach is a structure_ based one and thus has the advantage of being applicable to any type of graphs.

Our contributions in this paper lie in proposing a new method of filtering a graph based upon existing approaches: describing two filtering modes, and conducting experiments.

2 NodeRank: Node Importance Rank

Graph filtering is the selection of nodes and edges in a graph according to whether their attribute variables are within a specified range. The key question to be addressed in relation to filtering is determining which attribute(s) qualify for distinguishing nodes and edges in a graph. A node has many attributes such as the importance of its "role" in the graph, the content of what the node represents, the size of the node, and so on. As we know, nodes are employed to represent objects or entities, and edges represent relationships between these

objects or entities in graph visualization. In other words, the purpose of the use of a graph is to visually represent complex relationships between objects or entities in order to reduce the cognitive load. However, various nodes and edges in a graph play different roles in revealing such relationships. Some are prominent while others are trivial. Prominent nodes are those that are extensively involved in the relationships with other nodes. This involvement makes them more visible to the others. For this reason, it is necessary to develop a method to clearly identify the "important" nodes in a graph. A function is needed to measure the importance roles of nodes with respect to the depiction of relationships. Fundamental to our approach is the notion of *NodeRank*, which is defined as follows:

Definition 1 Node Importance Score (*NodeRank*): A real number indicating the degree of the important role in which a node plays in a graph, or of the involvement of a node in other relationships. Let $G = (V, E)$ be an undirected and connected graph, and R be a function that assigns a real value ranging from 0 to 1 to every node u in G. That is, $R: V \rightarrow [0,1]$. $R(u)$ called the *Node Importance Score* of node u ($0 \le R(u) \le 1$), is used to rank nodes.

Fortunately, the measure of "centrality" used in social network analysis [Faust 1995] provides the basis for the calculation of *NodeRank*. The concept of centrality seeks to quantify an individual actor's prominence within a network by summarizing structural relationships among the nodes.

Similarly, we develop a method to measure the importance of the nodes by means of their status or rank within a graph. In the following section, two approaches to calculating the *NodeRank* score are presented. In Section 3 two models of filtering graphs are given, followed by presenting the algorithm. Several examples are described in Section 5, and conclusion is given in Section 6.

2.1 Degree, Closeness and Between Centrality Index

As mentioned in Definition 1, a function R is to be constructed in order to map every node in a graph into a real value which can indicate its importance role within the graph.

Centrality in a social network refers to the importance of a particular node in a network. Measures of centrality have been developed to "attempt to describe and measure properties of 'actor location' in a social network" [Faust 1995]. The importance of a node in a graph can be measured both by how easily it reaches other nodes and how often it is directly connected to other nodes in the graph. In other words, The *Node Importance Score* of a node is determined by how it directly and indirectly connects to other nodes in a graph. In this view, the role of a node can be a function of its position in a given graph.

There currently exist a variety of centrality measures, but the main ones can be roughly classified into three major types: degree, closeness, and betweenness centrality [Faust. 1994, Freeman 1978, Friedkin 1991, Granovetter 1973, Katz 1953, Scott 2000], which are defined as follows, respectively.

Definition 2 ([Freeman 1978]) *Degree Centrality*: The number of edges attached to a node *u*.

Obviously, Degree Centrality can be normalized to range from 0 to 1, where 0 means the smallest possible and value 1 the highest possible centrality. The normalized measures are called relative measures of centrality:

$$C_D(u) = \frac{\deg(u)}{n-1}$$

where $\deg(u) = |\{v \mid (u,v) \in E \ \wedge \ v \in V\}|$, and $n=|V|$.

Degree Centrality reflects the direct relationships of a node with others in a graph. In other words, it indicates the number of links connecting adjacent nodes to a local node. So it is a local centrality.

Definition 3 ([Freeman 1978]) *Closeness Centrality:* The sum of geodesic distances, defined as the shortest path connecting two nodes, between a node *u* and all the other nodes.

$$C_C(u) = \frac{\sum_{v \in V} d(u,v)}{n-1}$$

where $d(u,v)$ is the shortest path between nodes u and v , which is equal to the number of edges between them. The measure of centrality based on closeness reflects a node's freedom from the controlling relationships of others, and its capacity for independent relationships within a graph. This measure actually indicates how far a node is from all others. A node with a higher closeness score is less centralized one than a node with a lower closeness score. The most central nodes can quickly interact with all the other nodes because they are close to all the others.

Definition 4 ([Freeman 1978]) *Betweenness Centrality:* The ratio of the number of the shortest paths between two nodes passing a node *u* to the number of all possible such shortest paths in a graph :

$$C_B'(u) = \sum_{j=1}^{n} \sum_{k=1}^{j-1} \frac{g_{jk}(u)}{g_{jk}}, \quad C_B(u) = \frac{C_B'(u)}{(n-1)(n-2)/2}$$

where g_{jk} is the number of the shortest paths from node j to node k, and $g_{jk}(u)$ is the number of these shortest paths that include node *u*.

Betweenness Centrality reflects the intermediate location of a node along indirect relationships linking other nodes. It "measures the extent to which a particular point lies between the various other points in a graph: a point of relatively low degree may play an important 'intermediary' role and so be very central to the network" [Scott 2000]. A node with high betweenness has the

capacity to facilitate or limit interaction between the nodes it links.

Both Closeness and Betweenness centralities are global. Note that all the above measures are relative ones, with the corresponding possible maximum values as denominators in the equations.

We observe that these three centrality measures may produce contrary results for the same graph. It can be a case in which a node has a low degree centrality, with a high betweenness centrality. Freeman [Freeman 1978] demonstrated that the betweenness centralities best "capture" the essence of important nodes in a graph, and generate the largest node variances, while degree centralities appear to produce the smallest node variances. In order to overcome the drawbacks of single centrality, the combination of degree, closeness and betweenness centralities yields the following measure as the *NodeRank* score:

$$R_n(u) = w_1 C_D(u) + w_2 C_C(u) + w_3 C_B(u)$$
$$= w_1 \frac{\deg(u)}{n-1} + w_2 \frac{\sum_{v \in V} d(u,v)}{n-1} + w_3 \frac{\sum_{j=1}^{n} \sum_{k=1}^{j-1} g_{jk}(u) \Big/ g_{jk}}{(n-1)(n-2)/2} \quad (1)$$

where the weights w_1, w_2, and w_3 sum to 1. For simplicity, we can give equal importance to the three measures by assigning equal values of the weights in our experiments. By adjusting the values of the weights, the emphasis can be placed on different measures. Generally, a node will have a high *NodeRank* score, if it has a high degree, is easily accessible to (close to) all other nodes, and lies on several geodesics (shortest paths) between other nodes.

The time complexity of these measures is obviously $O(n^2)$, meaning that the *NodeRank*s for large graphs should be computed offline.

2.2 Eigenvector Centrality

Degree, Closeness and Betweenness centralities are derived from graph distances. Nodes are judged to be important by how close or proximate other nodes in a graph are to them. However, we should simultaneously consider the importance of nodes that are proximate to the node under study. Being chosen by a popular individual should add more to one's popularity. Being nominated as powerful by someone who is seen by others as powerful should contribute more to one's perceived power. If one's influence domain is full of prestigious nodes, one's prestige should also be high. If, however, a node's domain contains only peripheral or marginally important nodes, then the rank of this node should be low.

Therefore, we should "weight" the distances used in the proximity indices by incorporating the importance of nodes in the influence domain.

- The *NodeRank* of each node is proportional to the sum of the *NodeRank* of the nodes to which it is directly connected.

$$r_i = \lambda \sum_{j=1}^{n} a_{ij} r_j$$

Where r_i is the *NodeRank* of node *i*, and a_{ij} =1 if node *i* and node *j* are adjacent, or 0 otherwise. Hence, a node connected to many nodes that are well-connected is assigned a high score by this measure. However, a node that is connected only to near isolates is not assigned a high score, even if it has a high degree.

On the other hand, some status measures of nodes do not completely depend on their connection to others. For example, each student in a class has some popularity that depends on his or her external status characteristics. In a communication network, each individual has sources of information that are independent of other group members.

- Each individual node has its own status characteristics, which are independent of other nodes.

Overall, the measure of the importance of a node depends on two factors: external and internal. The above equation can thus be modified:

$$r_i = \lambda \sum_{j=1}^{n} a_{ij} r_j + e_i$$

Where e_i is the internal importance score of node *i*.

These *n* equations can be rewritten in a matrix form. Let $\boldsymbol{r}$ be a vector of *Node Importance Scores*, i.e. $\boldsymbol{r} = (r_1, r_2, ..., r_n)'$, $\boldsymbol{e}$ a vector of the internal importance of nodes, i.e. $\boldsymbol{e} = (e_1, e_2, ..., e_n)'$, and $\boldsymbol{A}$ be an adjacent matrix[1]. An equation with a combination of both external and internal factors is then given by:

$$\boldsymbol{r} = \lambda \boldsymbol{A}^T \boldsymbol{r} + \boldsymbol{e}$$

The parameter λ reflects the relative importance of exogenous versus endogenous factors in the determination of importance scores. This equation has the matrix solution:

$$\boldsymbol{r} = (\boldsymbol{I} - \lambda \boldsymbol{A}^T)^{-1} \boldsymbol{e}$$

where $\boldsymbol{I}$ is the identity matrix of dimension *n*, and $\boldsymbol{e}$ is a vector of length *n*.

We normalize $\boldsymbol{r}$ to a length of 1, i.e.

$$|\boldsymbol{r}| = \sqrt{\sum_i r_i^2} = 1$$

The eigenvector importance of a node u_i is finally attained:

$$R(u_i) = r_i \quad (2)$$

Namely the components of vector $\boldsymbol{r}$ are the corresponding *NodeRanks* of nodes in a graph. High rank importance scores imply that nodes are connected either by a few other nodes that have high rank scores, or by many others with low to moderate rank scores.

[1] An adjacency matrix of a graph with *n* nodes is an *n* by *n* square matrix: $\boldsymbol{A} = (a_{ij})$, where a_{ij} =1, if node *i* and node *j* are adjacent, or 0 otherwise.

Eigenvector centrality is best understood as a variant of simple degree. Or it can be interpreted as a refined version of degree.

This idea of eigenvector centrality was initiated by [Katz 1953] and further developed by [Hubbell 1965] and many others, finally culminated with [Bonacich 1972] who defined centrality as the principal eigenvector of an adjacency matrix.

The nodes in a graph are ranked by their importance scores, and then all those nodes as well as their connected edges with their *Node Importance Scores* that are under a threshold are removed.

As stated before, filtering a graph aims to reduce the visual complexity. A filtered graph should still represent the main relationships between nodes after such a filtering. The basic requirement is that the filtered graph should be at least connected. The following definitions from graph theory [Harary 1972] are useful for keeping a graph connected.

Definition 5 A graph is connected if for every pair of nodes v_1 and v_2, there is a path starting at v_1 and ending at v_2.
A graph is connected if it can be travelled from any one node to any others along paths of edges. The graph is 2-connected if deletion of any node still keeps it connected; it is 3-connected if it still remains connected with removal of any two nodes, and so on. It is required that a *k*-connected graph have at least *k* + 1 nodes.

Definition 6 *CutPoint*: A node is a cutpoint if its removal disconnects a graph, i.e. increases the number of components. Also, its removal makes some points unreachable from some others.

The concept of a cutpoint can be extended from a single node to a set of nodes necessary to keep a graph connected. Those nodes are referred to as a cutset. A node cutset is a subset of the nodes of a graph, whose removal (simultaneously removing all edges adjacent to those nodes) makes the graph no longer connected. If the set is of size *k*, then it is called a *k*-node cut, denoted by $k(G)$. That is, the $|k(G)|$ of a graph is the minimum number of nodes that must be removed to make the graph *G* disconnected.

Definition 7 *Bridge***:** An edge is a bridge if its removal results in disconnected subgraphs. A bridge is an edge such that the graph containing the edge has fewer components than the subgraph that is obtained after the edge is removed.

Denoting the *Bridge* set as $\lambda(G)$, the edge connectivity $|\lambda(G)|$ of a graph is the minimum number of edges that must be removed to disconnect *G*.

Now we can formalize the requirements of filtering graphs. Let *F* be a set of the filtered nodes and associated edges, *t* a threshold of *NodeRank*, and G' the filtered graph, then the following properties of G' hold true:

$$1.G' = G - F$$
$$2.F = (\{v \mid v \in V \wedge v \notin k(G) \wedge R(v) < t\}, \{(v,u) \mid u \in V \wedge (v,u) \notin \lambda(G)\})$$
$$3.|G'| \leq |G|$$
$$4.G' \text{ is connected}$$

3 Filtering Graphs

There are two filtering models: Global filtering and Fisheye filtering. Both models are based on thresholds. However, Global filtering permanently removes relatively unimportant nodes (and their associated edges), measured by their *Node Importance Scores*. In other words, the appearance of a node is mainly determined by its *Node Importance Score*. In the Fisheye filtering model, whether a node should be visible or not is conditional on both the *Node Importance Score* of this node and its distance from the current focus node.

3.1 Global Filtering

With the ranked nodes, a filtered graph can be produced by suppressing all the nodes with their *Node Importance Scores* that are under a threshold specified and adjusted by users.
Suppose we denote the threshold as *t*, then a set of the remaining nodes after filtering is

$$V' = \{u \mid u \in V \wedge R(u) \geq t\}$$

where $R(u)$ is the *Node Importance Score* of node *u*.
An alternative means of obtaining the threshold is to specify what proportion of nodes will be reserved after filtering:

$$\frac{|V'|}{|V|} \leq t$$

The procedure is as follows: rank the nodes in decreasing order of their *Node Importance Scores*, and then sequentially remove the number of $(1\text{-}t)|V|$ nodes in order by starting from a node with the smallest score.

In order to measure the alteration of a graph after filtering, the density of a graph [Faust 1995] is employed:

$$\nabla(G) = \frac{2|E|}{|V|(|V|-1)} = \frac{\overline{Deg}(G)}{|V|-1}$$

where $\overline{Deg}(G) = 2|E|/|V|$ is the average degree of the nodes in the graph.

This is the proportion of possible edges that are actually present in a graph. In other words, the density of a graph is the average proportion of edges incident with nodes in the graph.

The additional properties of graph *G* and its filtered graph G' are as follows:

- $|V'| \leq |V|$ and $|E'| \leq |E|$
- $\nabla(G') \geq \nabla(G)$

A filtered graph does reduce the visual complexity, but still captures the main relationships between objects or entities in the original graph at the same time.

3.2 Fisheye View

Furnas [Furnas 1986, G.W.Furnas 1981] devised the fisheye lens model as an efficient way of showing large graphs. This model uses a "degree of interest" function to constrain (called Filtering fisheye views, FEV) or distort the display of information to relevant or interesting elements, according to their importance. The degree of interest (*DOI*) of a node *x* is a function of a priori importance (*API*) of *x* and the distance (*D*) between *x* and the focus node *f* [Furnas 1986, G.W.Furnas 1981]:

$$DOI(x) = API(x) - D(x,f)$$

It is clear that the degree of interest of a node increases with API and decreases with the distance. Generally, the importance and distance factors can be described as the size, and level in a hierarchy, or some other characteristics of objects in the collection. In the case of filtering a graph, *API(x)* is naturally determined by the importance of a node x, which is the exact implication of *Node Importance Score*. So we have:

$$API(u) = R(u)$$

As for the distance, *D(x,f)* is the graph-theoretic distance between a node *x* and the focus node v_f, i.e. $D(u,f) = d(u,v_f)$. Suppose *D* denotes the maximum distance over all pairs of nodes, which is the *diameter* of the graph $D = \max_{u,v_f \in V} d(u, v_f)$. The degree of interest can therefore be rewritten as:

$$DOI(u) = API(u) - D(u, v_f)$$
$$= R(u) - d(u,v_f)/D$$

With this normalization, the value of $DOI(u)$ ranges within (-1,1]. Given a node *u*, if $DOI(u) \geq t$, then the node *u* will be visible, otherwise invisible in the filtered graph, i.e. $V' = \{u \mid u \in V \wedge DOI(u) \geq t\}$ where *t* is the threshold. The effect of the filtering fisheye view is that nodes in the neighborhood areas of the focus node are shown in detail, while only the most important nodes in more distant areas are displayed, as determined by the above equation.

4 Algorithms

The algorithm for filtering a graph is as follows:

Input: A connected graph, a threshold
Output: A connected and filtered graph.

1. Compute the *Node Importance Score* of every node in a graph
2. Rank the nodes by their *Node Importance Scores*
3. Remove nodes and their associated edges which are not *Cutpoints* and *Bridges*, and whose *Node Importance Scores* are less than the threshold.

The above algorithm is founded on several basic algorithms reviewed here:
A graph is connected: A simple depth-first or breadth-first search suffices to test whether a graph is connected and to identify all the components in linear time. Specially, the algorithms for identifying nodes (or bridges) would try to delete nodes (or edges) one by one, and then use Depth-First Search (DFS) or Breadth-First search (BFS) to test whether the resulting graph is still connected. More complicated but linear-time algorithms exist for this problem, based on a depth-first search.

The shortest path between two nodes: There are two main algorithms for finding the shortest paths in a graph: Floyd's Algorithm and Dijkstra's Algorithm.

Eigenvector centrality: This can be computed as follows [Hotelling 1936, Loan 1996]:

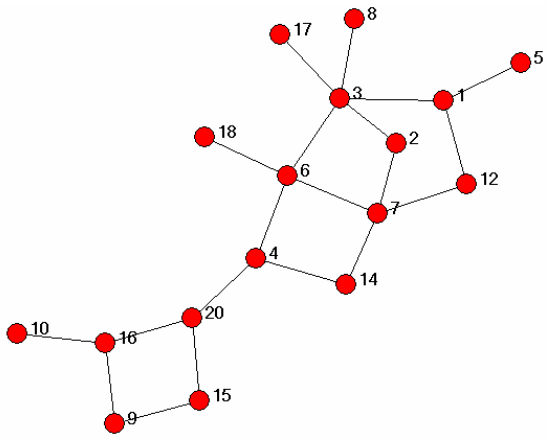
Step 0: Set $r_i = 1$, and assign initial values to e_i and λ, for all node *i*
Step 1: Compute $r_i^* = \lambda \sum_j a_{ij} r_j + e_i$
Step 2: Set γ equal to the square root of the sum of squares of r^* (the vector)
Step 3: Set $r_i = r_i^* / \gamma$ for all nodes *i*
Step 4: Repeat steps 1 to 3 until γ stops changing

Note that after executing step 1 the first time, *e** is equal to a simple degree.

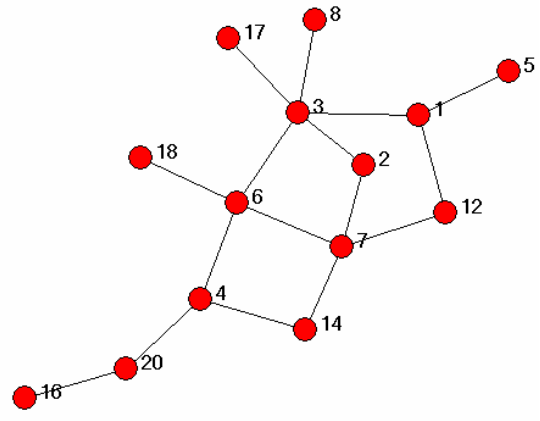
The running time for this computation comes mainly from the *Step* 1, i.e. $O(n^2)$. For a large graph, we can find the linkage of the graph offline, or gradually obtain part of the graph on-the-fly and then filter it.

5 Experimental Results

In this section several examples will be presented. The first simple example is a random graph with 17 nodes, and a density of 0.0735, in Figure 1(a). Figure 1(b) is the filtered version of this graph, with 14 nodes and a density of 0.0879. While the number of nodes in the filtered graph decreases from 17 to 14 due to the increased threshold, its density increases by 0.0144. This demonstrates that the filtered graph can still capture the main relationships represented in the original graph.



(a) A random graph with 17 nodes and Density 0.0735



(b) The filtered graph with 14 nodes and Density 0.1047

Figure 1: A random graph and its filtered graph

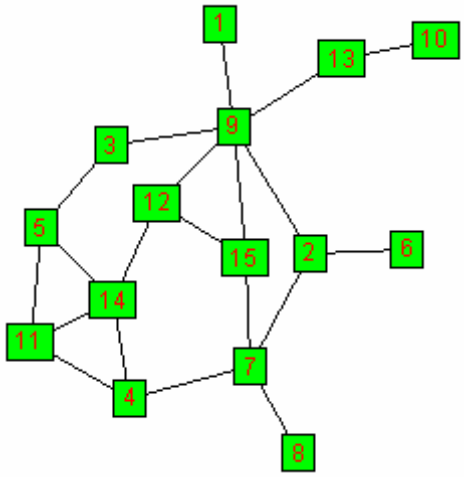


Figure 2: The structure of a small document collection on *Recipes*

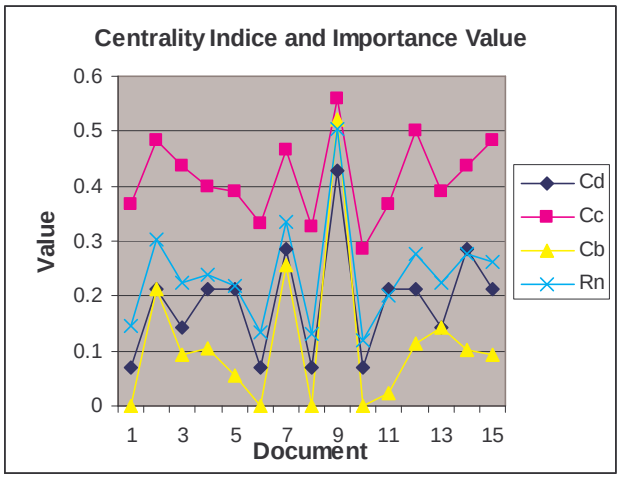


Figure 3: Comparison of centrality indices and *NoteRank* values of documents on *recipes*.

5.1 A Small Collection of "recipe" Documents

In this example we analyse the structure of a small collection on *recipe* documents by using *NodeRank*. Web pages and hyperlinks of this collection were gathered using the Web crawling software named *WebCrawler*. The result of a layout is shown in Figure 2 where nodes represent Web page documents, and edges indicate hyperlinks among the documents.

Centrality indices and *Node Importance Scores* of the collection are calculated. Figure 3 shows the comparison of the *Degree*(C_d), *Closeness*(C_c) and *Betweenness Centrality*(C_b), and *Node Importance Scores* (*Rn*) , the latter of which is calculated by formula (1). Compared with the other nodes, *Recipes* node (node 9) in the figure is obviously a prominent node with respect to those measures. Also, note that a different centrality index may lead to diverse interpretations. Node 1, for example, has the same value of *Closeness centrality* as that of node 11, but they vary greatly in their *Betweenness* centralities. As previously mentioned, different centrality measures focus on various aspects of the structure of a graph. For this reason, we use a combination of them in the equation (1) as the *Node Importance Score* instead of a single centrality measure.

In fact there are two kinds of documents in the collection: "hub" documents with many links, and "sink" documents with incoming links, but without out-going links [Kleinberg 1999, Lawrence Page 1998]. For instance, *Recipes* (node 9) and *Japanese Fried Rice* (node7) are "Hub" documents while *Numerical Recipes* (node1), *Catmeal cookies* (node 6), *Eggs pepper* (node 8) and *Main Dishes* (node 10) are "sink" documents. The importance of "Hub" documents surpasses other documents so that they have relatively high *Node Importance Scores.* A "Hub" is a transitional document through which users move to certain destinations while "sink" tends to be a final destination.

5.2 Web Graphs

Three examples of Web graphs will be presented in this section. In the first example, Huang's Web graph example [Huang, et al. 1998] was reconstructed in Figure 5 (a) and then was filtered shown in Figure 5(b).The *Node Importance Scores* of this exaple are illustrated in Figure 4. The two additional examples (Figures 6 and 7) were obtained as follows: first, use the *WebCrawler* with the starting URL addresses: http://www.it.swin.edu.au and http://www.unimelb.edu.au, and search depth 4, to produce a file, which includes all the URLs of the hyperlinked Web pages in part of the above Web sites, respectively. From this data collection, Web graphs are constructed where nodes represent Web pages, and edges represent hyperlinks among these pages. We next apply our proposed algorithm to filtering such Web graphs. Finally, the filtered graphs are laid out.

From these examples, we can conclude that: (1) the increase of the threshold results in the removal of more nodes and their associated edges. Equally well, the remaining nodes are those nodes with relatively high

Node Important Scores; or those that are adjacent to prominent nodes; (2) the densities of graphs increase as the sizes of graphs decrease.

6 Conclusion

Designing layout facilities for large graphs is one of the challenges in the field of graph layout. Examples of large graphs include the World Wide Web, communication networks, knowledge bases, semantic networks, and so on. In this paper, we have presented a novel method for filtering a graph. This method was motivated by the measure of the important role property of a node. The importance of a node in a graph is quantified as the degree to which it has direct and indirect relationships with others. In comparison with the rule based approach, our approach is a structure based one in that it makes use of the linkage of nodes instead of their semantics. So it is applicable to any kinds of connected graphs. The examples have demonstrated our approach can effectively reduce the number of nodes and edges, while the main linkage patterns in the original graph are still reserved. It can be concluded that Filtering is one efficient way to deal with large graphs. The future work will consider weighted graphs during filtering.

7 References

Bonacich, P. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, *2*. 113-120.

Faust, K., Stanley,W. *Social Network Analysis :Methods and Applications*. Cambridge University Press, 1995.

Faust., K, Stanley,W. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

Freeman, L.C. Centrality in social networks: I. Conceptual clarification. *Social Networks* (1). 215-239.

Friedkin, N.E. Theoretical foundations for centrality measures. *American Journal of Sociology*, *96*. 1478-1504.

Furnas, G.W., Generalized Fisheye Views. in *Human Factors in Computing Systems CHI '86*, (1986), ACM Press, 16-23.

G.W.Furnas. The FISHEYE View: A New Look at Structured Files, Bell Laboratories, Murray Hill, New Jersey, 1981.

Granovetter, M.S. The strength of weak ties. *American Journal of Sociology*, *78* (6). 1360-1380.

Harary, F. *Graph Theory*. Addison-Wesley, Reading, MA, USA, 1972.

Henry, T.R. Interactive Graph Layout: The Exploration of Large Graphs *Department of Computer Science*, University of Arizona, Tucson, 1992, 101.

Hotelling, H. Simplified calculation of principal components. *Psychometrika*, *1*. 27-35.

Huang, M.L., Eades, P. and Cohen, R.F., WebOFDAV – Navigating and Visualizing the Web On-line with Animated Context Swapping. in *7th World Wide Web Conference*, (Amsterdam, 1998), Elsevier Science, 636-638.

Hubbell, C.H. An input-output approach to clique identification. *Sociometry* (28). 377-399.

Katz, L. A new status index index derived from sociometric data analysis. *Psychometrika* (18). 39-43.

Kleinberg, J.M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, *46* (5). 604-632.

Lawrence Page, S.B., Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, Stanford University, 1998.

GH Golub and CF Van Loan, *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.

Scott, J. *Social Network Analysis: A Handbook*, London: Sage, 2000.

Shneiderman, B., The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. in *IEEE Conference on Visual Languages (VL'96)*, (Boulder, CO, 1996), IEEE CS Press, 336-343.

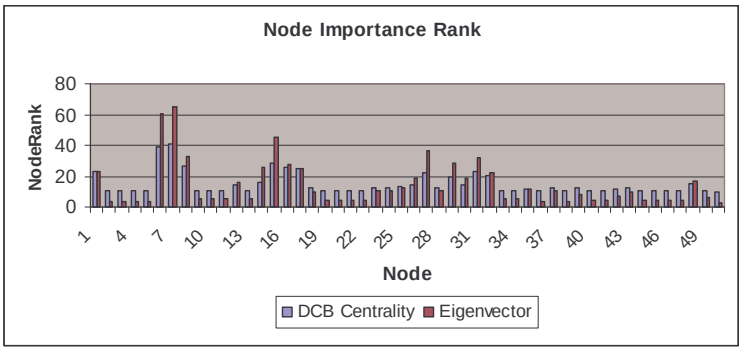
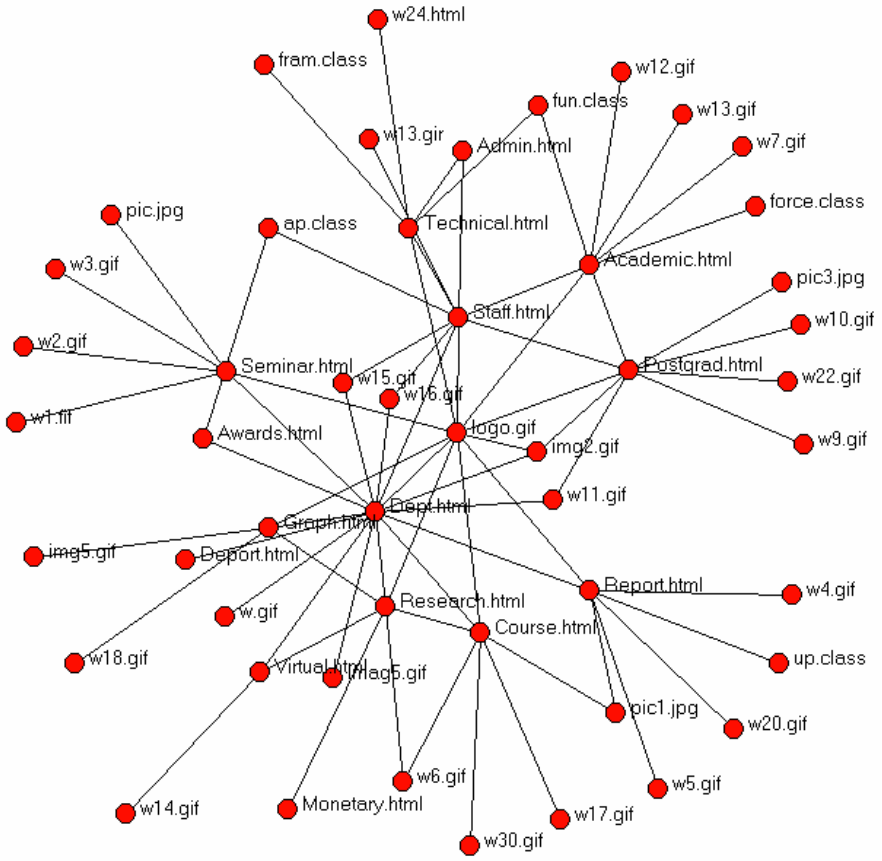
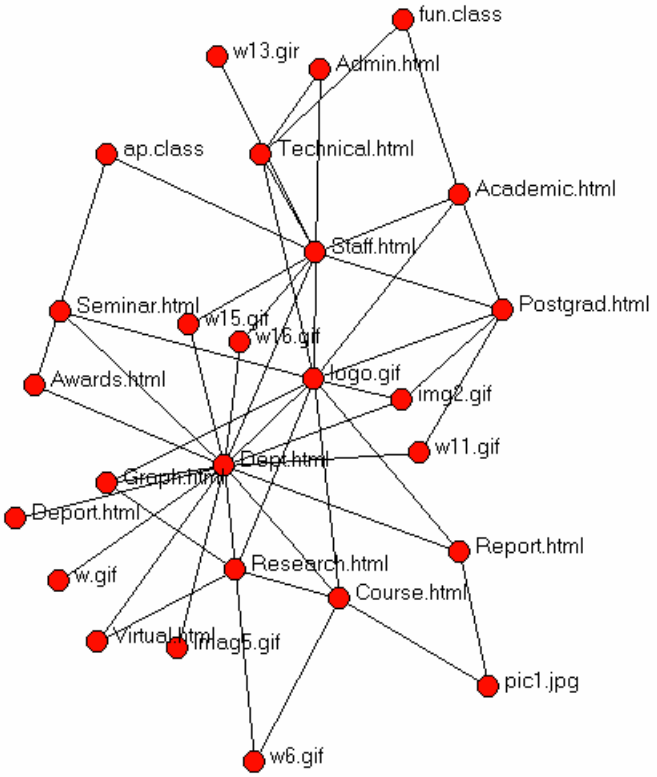


Figure 4: Comparisons of Node Importance Values in Figure 5(a) by DCB (formula 1) and Eigenvector(formula(2))



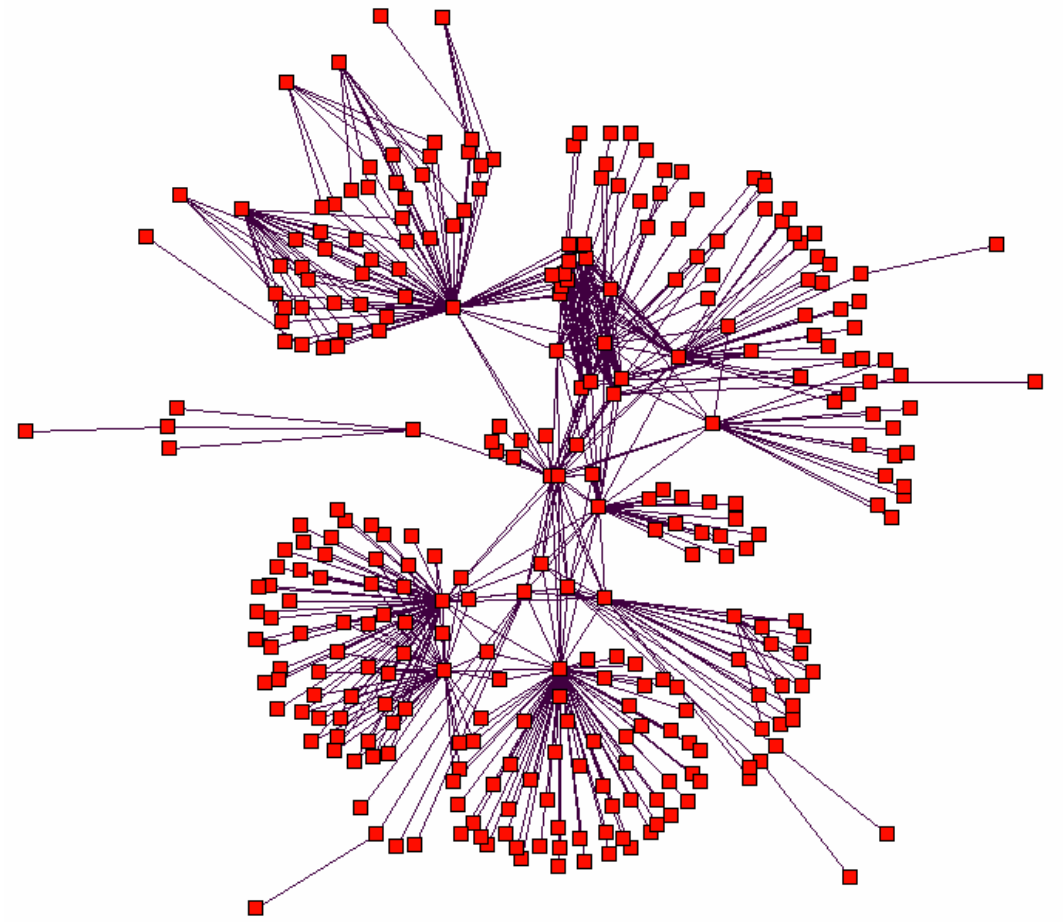
(a) A graph with 50 nodes, *NodeRank* range[2]: 2.874%~ 64.998 %**,** and Density 0.0380



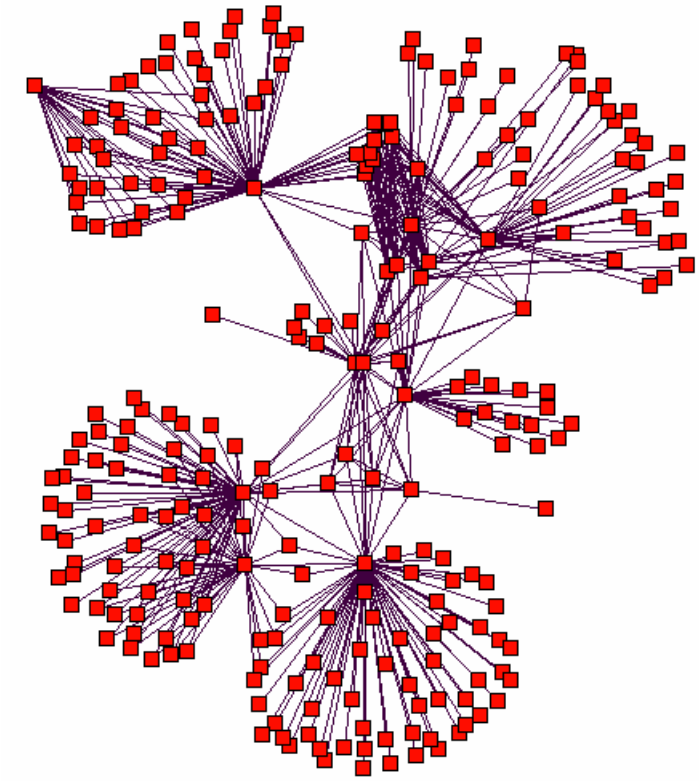
(b) A filtered graph with 27 nodes, *NodeRank* range: 7.474%~64.998 %, and Density 0.1062

Figure 5: A Web Graph

[2] The *Node Importance Scores* are calculated by Eigenvector, i.e., formula (2) , which also applies to the following examples.



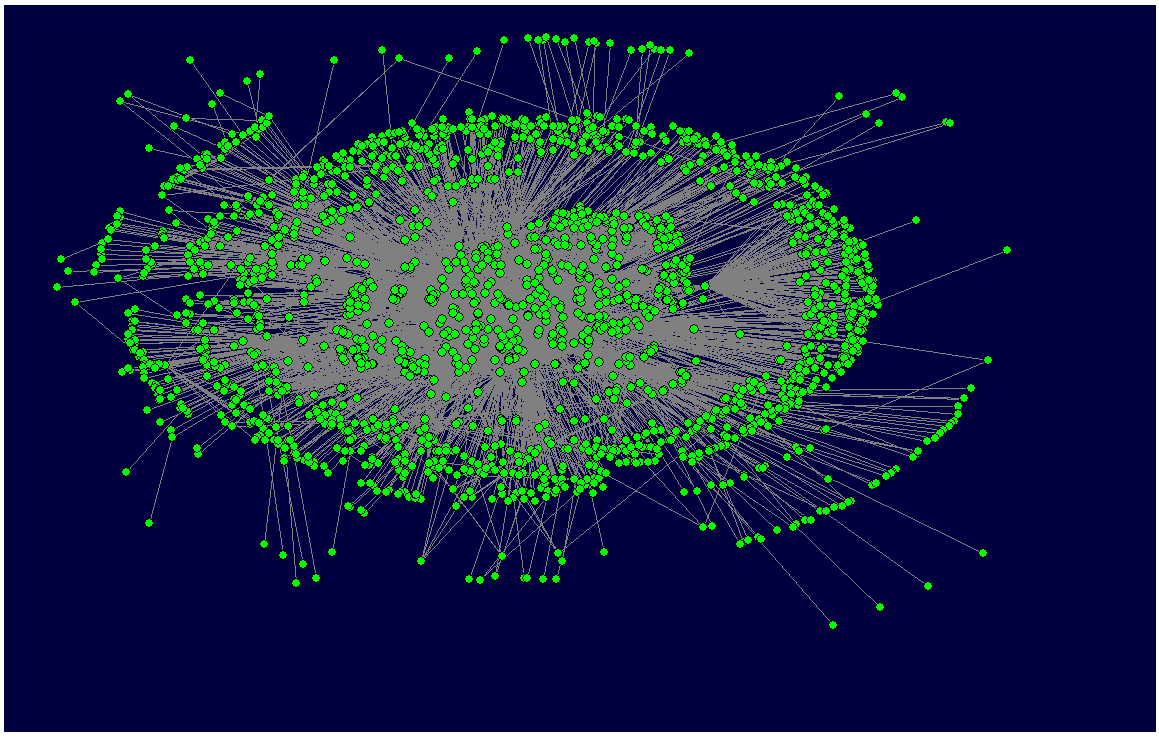
(a) A web graph with 317 nodes *NodeRank* range: 0.018%~37.438%, and Density 0.0055



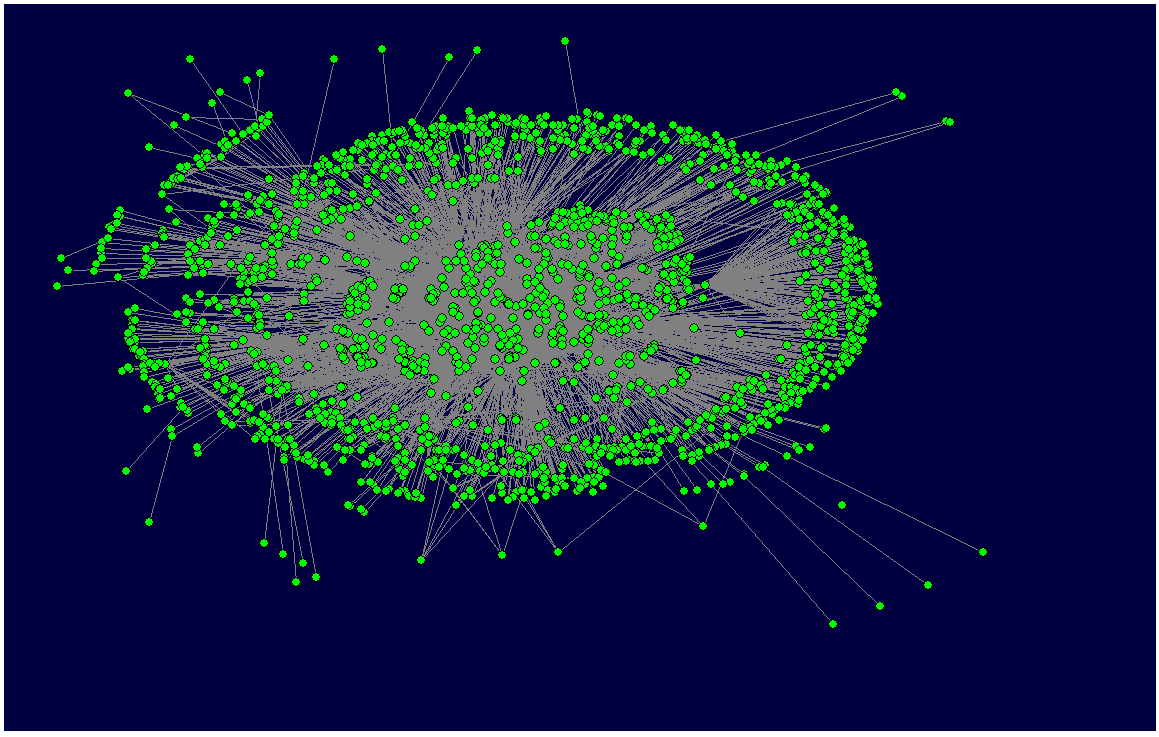
(b) A filtered web graph with 261 nodes, *NodeRank* range [3] : 2.086%~37.438%, and Density 0.0070

Figure 6: Part of Web Graph of Web Site of Swinburne University of Technology

[3] The threshold is 0.02



(a) A web graph with 2000 nodes, *NodeRank* range: 0.002%~27.549%, and Density 0.0016



(b) A filtered web graph with 1889 nodes, *NodeRank* range: 0.020%~27.549%, and Density 0.0017

Figure 7: Part of the Web Graph of the Web Site of the University of Melbourne