Effectively Visualizing Large Networks Through Sampling

Davood Rafiei*
Computing Science Department
University of Alberta

Stephen Curial†
Computing Science Department
University of Alberta

ABSTRACT

We study the problem of visualizing large networks and develop techniques for effectively abstracting a network and reducing the size to a level that can be clearly viewed. Our size reduction techniques are based on sampling, where only a sample instead of the full network is visualized. We propose a randomized notion of "focus" that specifies a part of the network and the degree to which it needs to be magnified. Visualizing a sample allows our method to overcome the scalability issues inherent in visualizing massive networks. We report some characteristics that frequently occur in large networks and the conditions under which they are preserved when sampling from a network. This can be useful in selecting a proper sampling scheme that yields a sample with similar characteristics as the original network. Our method is built on top of a relational database, thus it can be easily and efficiently implemented using any off-the-shelf database software. As a proof of concept, we implement our methods and report some of our experiments over the movie database and the connectivity graph of the Web.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques H3.3 [Information Storage and Retrieval]: Information Search and Retrieval

Keywords: visualizing the Web, large network visualization, network sampling

1 INTRODUCTION

The extensive growth of the Internet within the past few years has led to a proliferation of very large networks; examples include bibliographic collections, biological networks, market basket data, the Internet (both in the router and the inter-domain layers), and the World Wide Web. Although the collection and the storage of such data has become relatively straightforward, effectively analyzing data has proven to be more difficult. Visual display of networks, in particular, can lead to both better understanding and clear presentation of patterns that can often be hidden [20]. Alfred Crosby, the historian, lists "visualization" as one of the two processes that has led to the explosive growth of modern science; the other process is "measurement" [8]. Visualizing "large" networks, however, can be quite challenging if not impossible. This is due to the limitations of the screen, the complexity of layout algorithms and the limitations of human visual perception. Assuming that the network fits in memory, a good layout algorithm (eg. Force directed layout) can easily take quadratic time (in the number of nodes) for a single iteration. Often you need multiple iterations, which may depend on the number of nodes in the graph, to achieve a good result. The graph structure of the Web, for instance, is far too large to hold in the memory of most desktops let alone visualize it.

To gain insight into the complexity of the problem, consider the graph structure of the Web at the domain level as shown in Figure 1-a using the GEM layout algorithm [12], Figure 1-b using the GEM3D layout algorithm, and Figure 1-c using a force directed layout algorithm [10] [1]. This network is relatively small, having only 224 nodes and 8790 edges, but it is still not easy to find any interesting patterns. This is by no means a limitation of a layout algorithm but illustrates the complexity of visualizing large general graphs. Scaling up the visualization to a graph of the Web with millions of nodes at the site level or hundreds of millions of nodes at the page level is quite challenging if not impossible.

Our proposed alternative is to refrain from visualizing the entire network. At the core of our methods is sampling. We sample the network and only visualize the sample. Even though the network can be quite large, the size of the sample can be adjusted to match the limitations of the visualization environment. We study some of the topological properties of a network that are preserved in a sample and show that a relatively small sample, if collected carefully, can still show some of the patterns that are inherent in the entire network.

As our second contribution, we develop a notion of "focus", one can set, to bring into focus only part of the network that needs to be explored in greater detail. This is done in the context of the full network. If no focal point is set, the network is sampled uniformly. In the presence of a focal point, the sampling is biased toward that focal point, thus the visualization emphasizes the focal point and its neighbourhood in the network.

In this paper, we propose several sampling-based schemes for both focusing the search and visualizing networks which are too big to be fully visualized. We formalize a notion of focus for both networks with directed and undirected graph structures. We further extend this formulation to the case where edges in the underlying graph structure are weighted.

We build a prototype, named *ALVIN* [2], that implements the ideas described in this paper, and run it over the connectivity graph of the Web. We abstract the Web graph into three layers: *domain*, *site* and *page*, and run experiments over these layers. Some of our experimental results and other anecdotal evidence are provided to show that our proposed schemes can be quite useful.

The rest of the paper is organized as follows: we discuss issues related to sampling a network in Section 2. Our proposed scheme for visualizing and expanding a network is discussed in Section 3, and our notion of focus is presented in Section 4. Section 5 presents some implementation details and our experimental results. Section 6 reviews the related work, and Section 7 concludes the paper.

2 SIMPLE RANDOM SAMPLING OF A NETWORK

In this section, we discuss several ways of sampling a network and some of the characteristics of the original network that can be observed in the sample. In the next section, we formalize these sampling schemes in the form of some growth processes and develop a general model for visualizing a network.

*email:drafiei@cs.ualberta.ca
†email:curial@cs.ualberta.ca

[1] The GEM and GEM3D algorithms were implemented in Tulip [32]. For a force-directed layout, we used the spring layout algorithm, implemented in LEDA [22].

[2] The name *ALVIN* stands for *Alberta system for Visualizing Large Networks*.

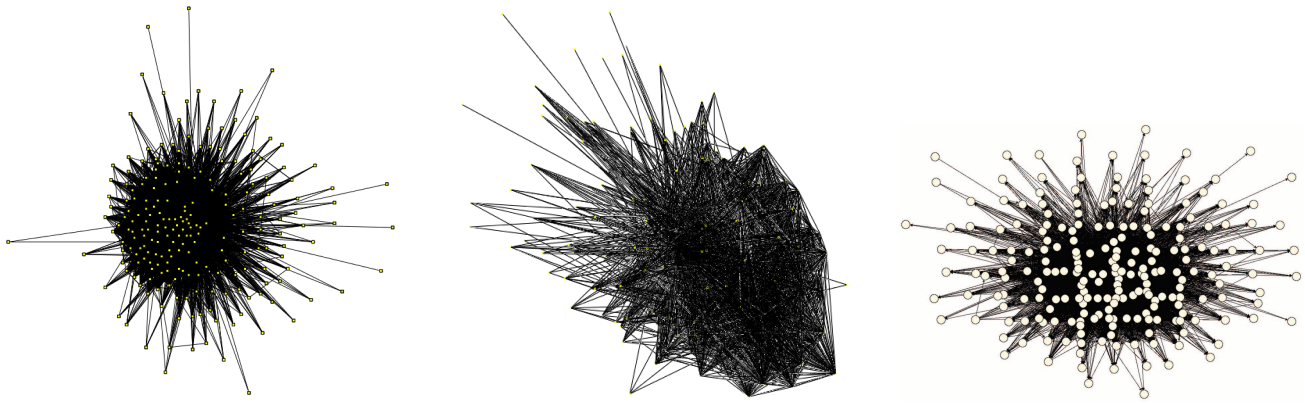

Figure 1: *The connectivity network of the World Wide Web at the domain level. (a) Frick's GEM layout algorithm is used, (b) GEM3D is used as the layout algorithm, and (c) a spring layout algorithm is used.*

Given a network $G(V_G, E_G)$, any subgraph of G can be treated as a sample of the network. Clearly, there are different ways of taking a subgraph and as a result there are many different sampling strategies. We use the following three methods for obtaining a simple random sample of a network. Independent of the strategy used, we let $S(V_S, E_S)$ denote a simple random sample of G.

SRS_1: Take a simple random sample of the nodes, V_S, and let $S(V_S, E_S)$ be the subgraph of G induced by V_S (i.e. the subset of the vertices, V_S, and edges with both endpoints in this subset).

SRS_2: Take a simple random sample of the edges, E_S, and let $V_S \subset V_G$ be the set of nodes that are incident to at least one edge in E_S.

SRS_3: Take a simple random sample $S'(V'_S, E'_S)$ using SRS_2 and let $S(V_S, E_S)$ be the subgraph of G induced by the nodes in V'_S.

Given a network of N nodes, SRS_1 randomly selects n nodes and includes all the edges between the selected nodes. Thus a node of degree k is expected to have a degree of $k\frac{n}{N}$ in the sample. The caveat is that this number is expected to be almost zero unless the sample includes a large fraction of the nodes, k is large or both.

SRS_2 may be more desirable when the sample size is small or the network is not well-connected. This sampling is unbiased toward edges but not toward nodes. Nodes with large in- or out-degrees are more likely to be in the sample, and paths of length greater than one are likely to form between them. This is not as problematic as it may look since those nodes are likely to form the backbone of the network and it is good to have them in the sample.

SRS_3 bears similarities to both SRS_1 and SRS_2. As in SRS_2, nodes with large degrees are more likely to be included in the sample. However, like SRS_1, there is a direct relationship between the degree of a node in the sample and its actual degree.

Other strategies for sampling a network are also feasible (e.g. see [21]) and can be used with our framework.

2.1 Using Sampling to Visualize Network Topology

There are a number of traits which are found in every network, and can be useful in describing the general topology of a network. These include degree distribution, connected component size distribution, average path length, clustering coefficient, etc. Some of these traits can be preserved when sampling from a network. The degree distribution of the Web graph, in particular, provides stratified counts of the degrees, differentiating hubs and authorities from other pages [6]. This property in turn can be useful in a visualization, as evidenced by some of our examples. The component size distribution is another important visual feature of a network (e.g. see the results reported for the Web graph [9]), and we often want it to be preserved in a sample. We discuss these features in the context of the movie database from *IMDb* [3], where each actor is represented by a vertex and there is an undirected edge between two actors if the actors are cast together in the same movie.

2.1.1 Path Length and Clustering Coefficient

A class of networks, known as small-world networks, are characterized by a short average path length and high clustering coefficient[4]. It is not clear if the average path length and the clustering coefficients can be predicted from a sample. Consider G_1 as a complete graph and G_2 as a complete bipartite graph. The clustering coefficients of G_1 is 1 and G_2 is 0. For a small sample of both graphs taken using SRS_2, the clustering coefficient may not be well-defined if the selected edges or nodes are not connected. After increasing the sample size, the clustering coefficient for G_1 can be either undefined, if the sample is not connected, or any number in the range [0,1]. In the latter case, the clustering coefficient is dependent on the selected edges and does not monotonically change with the sample size. Similarly, the clustering coefficient for a sample of G_2 will be either undefined or zero again depending on the edges that are selected but not the sample size. Finding a sampling strategy that can provide an estimate of the clustering coefficient for a general graph is an open problem. The average path length, on the other hand, is only defined between nodes that are connected. Two nodes that have a path between them in a parent graph may not be connected in a sample. This makes it difficult to infer a correlation between the average path length of a sample and that of its parent graph in general, but there are more specific cases where a correlation exists (e.g. when SRS_1 is used for sampling).

2.1.2 Degree Distribution

Figures 2, 3 and 4 show the degree distributions respectively using SRS_1, SRS_2 and SRS_3 for sampling from the movie database; the sample size is varied from 5% to 100%. For SRS_1 and SRS_3, there is

[3] *IMDb* - Internet Movie Database (www.imdb.com)

[4] The clustering coefficient gives the average degree to which vertices adjacent to a node are adjacent to each other. See Watts[33] for definitions.

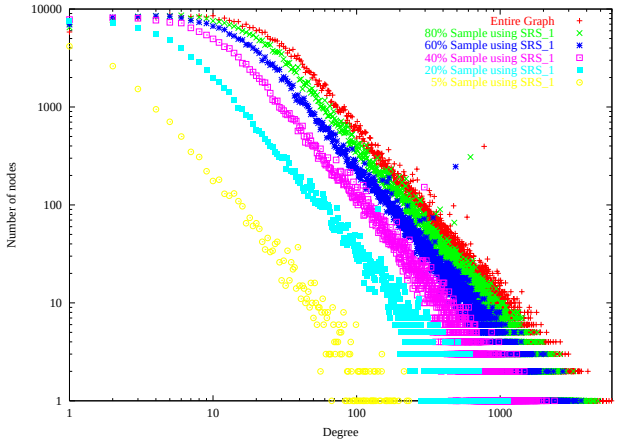

Figure 2: *Degree distribution using SRS_1 for sampling.*

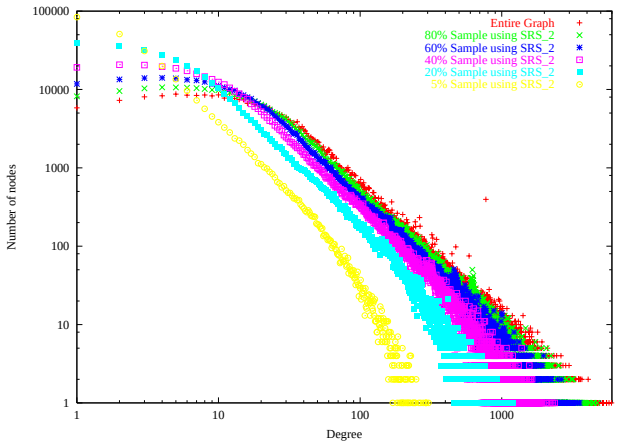

Figure 3: *Degree distribution using SRS_2 for sampling.*

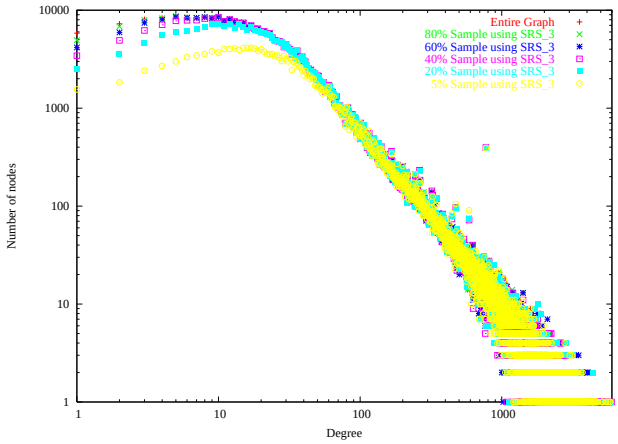

Figure 4: *Degree distribution using SRS_3 for sampling.*

a direct relationship between the actual degree of a node and its degree in a sample. This is shown by a relatively constant gap between the actual and estimated distributions; the gap becomes smaller for larger samples. For SRS_2, our samples overestimate the real distribution for smaller degrees and underestimate the real distribution for larger degrees. This is consistent among all samples. As a result, the degree distribution of even a 5% sample for degrees 4, 5 and 6 is very close if not the same as the actual distribution. Overall, the gap between the actual and the estimated distributions using SRS_2 is smaller than that of SRS_1, and the gap between the actual and estimated distributions using SRS_3 is smaller than that of SRS_2. Our estimates using SRS_3 must be treated with care. The depicted sample sizes show the fraction of edges selected in the first step of the sample only. The size of the sample after adding all edges between the selected nodes in the second step can be larger and can vary from one network to another. For instance, a 5% sample in one case includes over 50% of the edges. This explains why the estimated and the actual distributions are very close.

2.1.3 Component Size Distribution

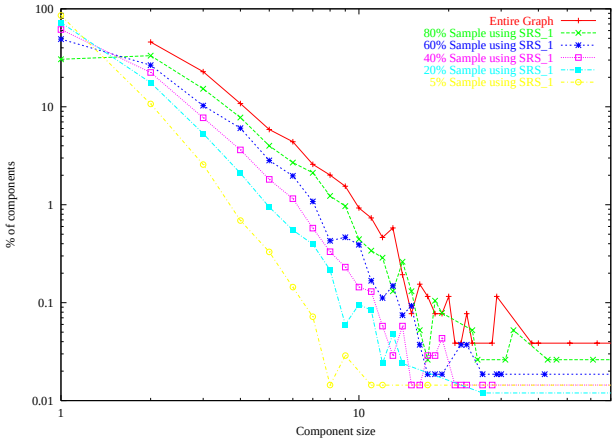

Figure 5: *Connected component size distribution for SRS_1*

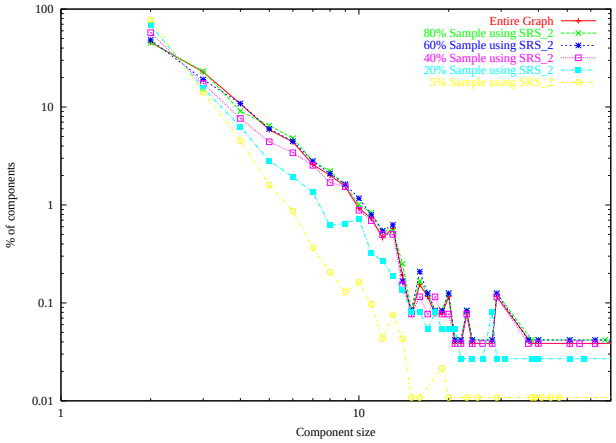

Figure 6: *Connected component size distribution for SRS_2*

Figures 5, 6 and7 show the component size distributions of the movie database, respectively using strategies SRS_1, SRS_2 and SRS_3

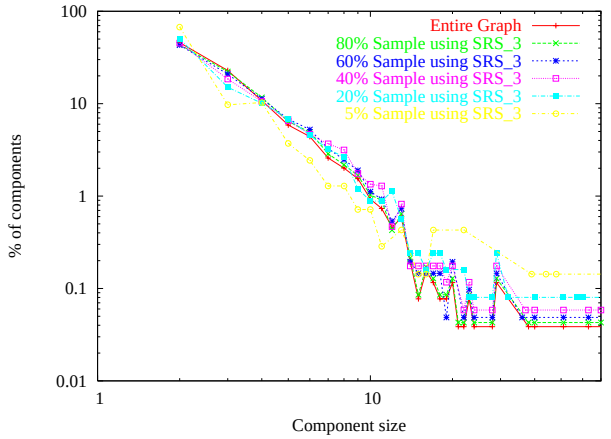

Figure 7: *Connected component size distribution for SRS_3*

for sampling. The sample size in both graphs is varied from 5% to 100%. A small and a consistent gap between the distributions of a sample and the entire graph indicate that the sample closely resembles the original data.

There have been some attempts to find a relationship between the distribution of component sizes in a sample and the number of components in the entire network. For transitive graphs [5], in particular, Frank has shown that if the network is sampled using SRS_1, the resulting network can be used to find an unbiased estimate of the number of connected components of the entire network [11].

Theorem 1. *Let the parent graph be transitive, and suppose $S(V_S, E_S)$, a simple random sample taken using SRS_1. Let $v = |V_S|$. If $K_r(S)$ denotes the number of connected components of size r in the sample, then an unbiased estimate of the number of connected components in the parent graph is given by*

$$\sum_{r=1}^{M} (1 - C_r) K_r(S)$$

where

$$C_r = (-1)^r \binom{N - v + r - 1}{r} \binom{v}{r}^{-1},$$

N is the number of nodes in the parent graph, $M \leq v$ is a constant and the parent graph has no connected component of size larger than M.

Both the proof and the variance of this estimate is given by Frank [11]. In practice, we are often interested in visualizing graphs which are not transitive, hence this theorem is not directly applicable. However, there are some indications that the estimates may still be a useful approximation for other graphs. Abusing the theorem, we tried using SRS_2 with Frank's estimate on synthetic data. Our synthetic data included graphs consisting of both complete connected and complete bipartite components. The component sizes were generated randomly and varied from 4 to 80. The results showed that Frank's estimate used with SRS_2, sampling only 25% of the edges, could accurately estimate the number of components with an average error of less than 8%.

[5] A graph is *transitive* if there is an edge between every connected pair of vertices.

3 Network Growth

If the network that is being visualized is too large, it may not be feasible to preserve some of the desired topological properties of the network in a small sample. To address this problem and to provide a navigation scheme, we develop several growth processes, collectively referred to as *network growth*, that allows one to interactively visualize a network.

In an interactive fashion to some degree similar to Web browsers, a visualization may start with a small subset of the network which may include a set of hand-picked nodes and edges or the result of a query. This is useful for narrowing down the visualization to some of the interesting elements when the network is too large to be fully visualized. The visualization may proceed towards the goal by iteratively growing the initial set. A novelty of our method is that some user-controllable parameters describe how and to what degree the network must be expanded. The expanded network often has more detail about the elements being studied yet is small enough to be visualized and internalized. After a few layers of expansion, the network may become too large; this may be an indication that the browsing should switch to another small subset.

Let $G(V_G, E_G)$ be the network that needs to be visualized and $C(V_C, E_C)$, a subgraph of G, be the network that is currently displayed on canvas. We want to select nodes from $V_G - V_C$ and edges from $E_G - E_C$ and add them to C, thus expanding the network on canvas with respect to G. Next, we discuss several ways of expanding a a network, formalizing our earlier sampling schemes in the form of some growth processes.

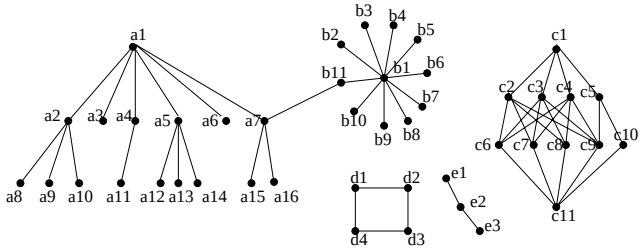

Figure 8: *A network instance.*

3.1 Global Growth

Sometimes we want to gain some insight into the general connectivity structure of the network without specifying a pivotal point; or we might be interested in only part of the network but want to browse this part in the context of the entire network. We may achieve this by taking a simple random sample of the network and visualize the sample. One such sample can provide the general connectivity structure of the network and maybe some common patterns without emphasizing one specific part. Clearly, the larger the sample, the more accurate the estimates; though a detailed sample may not always be clearly visualized.

Definition 1. Let C be a subgraph of a parent network G. A *global growth* of C with respect to G adds to C a simple random sample of G taken using one of the sampling strategies from Section 2.

Example 3.1. Figure 8 shows an instance of a network with 45 nodes and 55 edges. A simple random sample obtained using SRS_2, with only six edges picked from the random ordering shown in the appendix is displayed in Figure 9-a. This sample, consisting of 11% of the edges, shows some of the components of the parent network; it has the same number of connected components as the

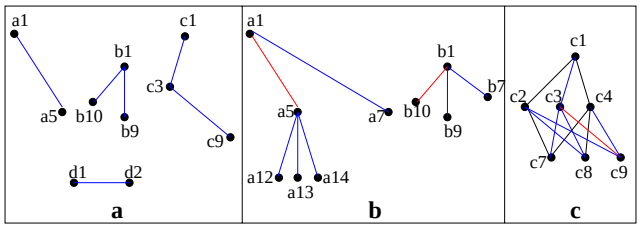

Figure 9: **(a)** *a global growth,* **(b)** *a local growth with initial edges (a1,a5), (b1,b10) and* **(c)** *a local growth with initial edge (c3,c9).*

parent network even though the components are not necessarily the same.

3.2 Local Growth

We often know some of the nodes and maybe some of the edges of a network and wish to find more nodes and edges that are somehow related to our starting set, or we may want to find out how our starting set is perceived within the general structure of the entire network. This can be done through sampling from the network surrounding C and adding the sample to the canvas. The sample includes some of the edges that glue C to the rest of the network G.

Definition 2. Let C be a subgraph of a parent network G, and let $I(V_I, E_I)$ be the subgraph of G such that E_I is the set of edges with one endpoint in V_C and the other endpoint in $V_G - V_C$ and V_I is the set of nodes incident to any edge in E_I. A *local growth* of C with respect to G adds to C a simple random sample of I taken using one of the strategies from Section 2.

Our local growth generalizes a sampling method, often referred to as *snowball sampling*, which is typical of a link-tracing design where a simple random sample or stratified random sample of units is selected and all other units linked to the initial sample are included or observed [31]. Unlike a snowball sample, the initial set in a local growth is not necessarily picked randomly; instead, it can be the result of a user query. Furthermore, a local growth does not necessarily include all edges linked to the initial set since this can be too large.

Example 3.2. Figure 9-b shows the result of a local growth after hand-picking the edges $(a1, a5)$ and $(b1, b10)$ from the network in Fig 8 and adding 6 more edges selected using SRS_2 through a local growth. For our edge selection, we again use the random ordering in the appendix but only add edges with one endpoint in {a1,a5,b1,b10}. As another example, Figure 9-c shows the result after hand-picking $(c3, c9)$, doing a local growth using SRS_3 which adds 6 more edges (these edges are coloured blue) and further extending the graph to include edges with both endpoints already selected (these edges are coloured black). Compared to a global growth that shows more of the structure of the entire network with less resolution, a local growth depicts a specific part of the network in greater detail but with less information about the network as a whole.

3.3 Mixed Growth

A local growth can be combined with a global growth at a user-specified rate to provide a more balanced mixture of the two. Under this scheme, called a *mixed growth*, the network is sampled as follows: with some probability we perform a local growth and with the remaining probability we perform a global growth. A mixed growth provides a spectrum of sampling schemes with local and global growths at the two ends of the spectrum.

3.4 Wiring and Rewiring

Sometimes we have our desired nodes on canvas and wish to visualize the interconnections between them in greater detail. A solution is to add more edges between the nodes on canvas. This is a special case of SRS_2 where the sample is taken from the graph induced by the nodes on canvas; we call this process *wiring*. In the extreme case, a wiring can add all the edges between nodes on canvas. However, this may clutter the visualization, hence a user-specified parameter may be used control the degree of wiring.

Since selecting edges is a random event, there are many possible wirings, and we may wish to view more than one possible wiring of the nodes on canvas. Through the process of *rewiring*, all edges on canvas can be removed and the nodes on canvas can be wired again. This may reveal properties that may not have been displayed by the original wiring.

4 Focused Browsing

A network growth already provides a way to focus on a specific part of the network, but the part of the network we can focus on is always fixed to I in a local growth and to G in a global growth. In general, we may want to focus on a part of the network other than I and G. We introduce a more general notion of *focus* that can be used to narrow the visualization to a desired part of the network, reducing both the size and the complexity of the visualized network. Our notion of "focus", referred to here as *focal point*, formulates to some extent our *interest* in the network. Without loss of generality, our browsing goal is to visualize the focal point in the context of the entire network.

The following scenario shows how a focused browsing can be useful. Consider the connectivity graph of the Web where nodes represent Web pages and edges describe the hyperlinks between pages. Suppose we are interested in the connectivity of pages on a specific topic, say *surfing*. We can set the focal point to include all pages that mention the term 'surfing' in their contents. There can be many more pages on this topic than what we can fit on canvas, thus we may visualize only a subset of these pages. If we expand the visualized set by adding pages that either link to a page in the initial set or are linked by a page in the initial set, the resulting set is shown to include the most prominent sources of primary content known as *authorities* and high-quality guides and resource lists known as *hubs* on the search topic[19].

It is not hard to integrate this notion of "focus" into our visualization scheme. Since our visualization is based on sampling, in the presence of a focal point, the sampling is biased toward this focal point.

4.1 Formal Model

Given a network $G(V,E)$, a *focal point* is a subgraph $F(V_f, E_f)$ where $V_f \subseteq V$ and $E_f \subseteq E$. In the absence of a focal point, F is naturally G, meaning that we are interested in the entire network.

A *focused growth* describes how the network on canvas can be expanded using a sample of the parent network and with respect to a focal point F. Next, we define a focused growth which takes two real number parameters that control the degree of bias towards the focal point, and can be used with either SRS_2 or SRS_3. A similar definition can be formalized for SRS_1 which is not discussed here.

Definition 3. Let C denote the graph on canvas and F denote the focal point; both are subgraphs of a parent network G. Denote with E_I the set of edges that have one endpoint in F and the other endpoint not in F. Let the input parameter $r \in [0,1]$ denote the probability that one endpoint of a selected edge should be sampled from the focal point and the input parameter $s \in [0,1]$ denote the probability that the other endpoint should also be sampled from the focal point.

A *focused growth* at rate (r,s) of the network on canvas C with respect to the focal point F and the parent graph G adds to C simple random samples of E_I, E_G and E_F with sample sizes respectively proportional to $s(1-r)+r(1-s)$, $(1-r)(1-s)$ and rs (the nodes incident on sampled edges are obviously selected).

A focused growth combines two *simple random samples* with a *snowball sample* at a user-specified rate. If we set the focal point to the network on canvas and $r = 1$ and $s = 0$, a focused growth simulates the local growth of Section 3.2. A focused growth also simulates the global growth of Section 3.1 if we set the focal point again to the network on canvas and $r = 0$ and $s = 0$. Varying the values of the parameters r and s, we can obtain other variations of a network growth.

4.2 Directed and Weighted Networks

It is not hard to extend our proposed schemes to both directed and weighted networks. For a directed network, we may fix in advance the fractions at which a source and a destination must be selected from V_F. One simple setting, for instance, is to set the ratios to 50/50 or some other constant. An alternative is to allow the ratios to be set during browsing using additional parameters.

In a weighted network, often the weight of an edge describes the strength of the relationship between the two endpoints. In a commuting network, for instance, each edge may be weighted to indicate the frequency of travels made in a day. If the network consists of more than one level of abstraction, each node or each edge in a more general layer may be weighted and the weight may aggregate multiple nodes or edges from a more specific layer. For instance, the connectivity graphs of the Web on the domain and site levels can be seen as aggregations of the Web graph on the page level. If the weight of a node or an edge is treated as an indication of its importance, we want to bias the visualization towards highly-weighted edges. This is again possible within our sampling framework by replacing a simple random sample with a weighted sample.

5 Experiments

ALVIN, our current prototype implementing these ideas, has the following highlights:

- It uses the DB2 relational database as its back-end data storage and querying engine. It makes no assumption on the size of the network and the back-end relational database can efficiently handle very large data sets.
- It provides an interface for both focusing and expanding the network on canvas. It allows the user to interactively expand the graph on canvas using parameters r, s and *the size of the sample*. Requests that arise from user interactions are mapped to SQL statements and are directed to the back-end SQL engine for an efficient evaluation.
- It is developed in C++ using the LEDA class library [22] and makes use of the layout and graph algorithms that are available in this library.
- Network abstraction and hierarchical views are supported by creating tables and views in the relational database.

We ran ALVIN over the linkage structure of a snapshot of the Web from *Internet Archive* [6]. Each vertex in the dataset denoted a Web page and each directed edge denoted a hyperlink, and the network was stored as relational tables. We also constructed two hierarchical views of the data in the site and the domain levels. These graphs were weighted with the weight of an edge representing the number of links from one site (domain) to another. For efficiency reasons, these views were pre-computed and physically stored. Our Web graph was a snapshot taken in 1999 and included 178 million nodes and 800 million edges. Next, we report some of our results with this data set.

[6]The *Internet Archive* is a public nonprofit organization that offers access to historical collections that exist in digital format, including the snapshots of the Web (www.archive.org)

5.1 Bow-Tie Shapes

The Web has been shown to be made up of a few distinct components: (1) a core in which every node can be reached from every other node, (2) an 'IN' that includes the nodes that link to the core but are not linked back, (3) an 'OUT' that contains the nodes that can be reached from the core but do not link to the core, and (4) the remaining set of nodes which cannot reach or be reached from the core but are connected to 'IN', 'OUT', both or neither. These components are best represented using a bow-tie shape [6].

One question is if we can enumerate some of the instances of these components, in particular those that are relevant to our search. For instance, given a set of nodes, we may want to find other nodes which can form a bow-tie shape with our initial set. The result can give more details on the nature of the nodes in each component and the connections within and between the components. In an attempt to enumerate some of the members of these components, we started with three nodes which were believed to be in the core; these included `dir.yahoo.com`, `yahoo.com` and `yahoo.ca`. We did 10 local growths, each time adding 50 edges, followed by a focused growth, adding 100 edges with $p = q = 1$ and the focal point set to the network on canvas. SRS_2 was used for sampling. The result as shown in Figure 10 included more nodes in the core such as `sec.yahoo.com`, `anglefire.com` and `ncsa.uiuc.edu` interconnected with nodes in OUT such as `cfa-www.harvard.edu` and `newscientist.com` and nodes in IN such as `internetcollege.net` and `chess-space.com`.

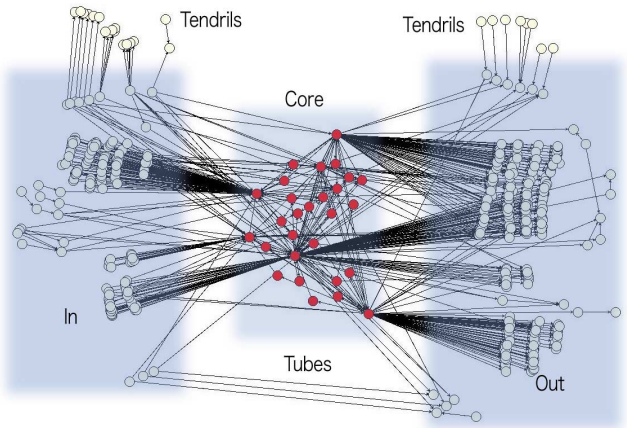

Figure 10: *Instances of the bow-tie shape in the Web graph*

5.2 Natural Clusters

Related pages on the Web often form clusters, for instance, if they are on the same topic. To visualize these clusters within our scheme, we started with a few nodes as our seed set. The seed set in one experiment included two news sites (`news.bbc.co.uk` and `www.foxnews.com`), two CS department

home pages (www.cs.wisc.edu and www.cs.cornell.edu) and the Science Magazine home page (www.sciencemag.org). We did a global growth, taking a 0.1% sample using SRS_1 and adding 168,000 nodes to our initial set. We expected that a dense connection between related pages will yield a high probability that these pages are part of the same connected component. For clarity, Figure 11-a shows only the components that include the pages in our seed set. Three clusters are formed around our seed set: one cluster includes the two CS departments with other CS departments such as www.lcs.mit.edu added through the sampling. Another cluster includes the two news sites, and the third cluster includes the Science Magazine. Comparing the degrees of the nodes in our seed set, the Science Magazine has a larger degree than the two news sites which in turn have larger degrees than the two CS departments. This is consistent with their actual degrees in the parent network.

To add more details of the interconnections and within a global growth, we took another 0.1% sample of the nodes using SRS_1. This combined with our earlier sample gave a 0.2% sample of the nodes. As shown in Figure 11-b, the nodes in our initial seed set now form a single connected component. It is easy to see that the average path between the news sites and the Science magazine page is shorter than the average path between the CS department home pages and the Science magazine page. Compared to the Fox News, the British Broadcasting Co. (BBC) has a shorter path to the Science magazine, possibly due to its coverage of the science articles.

Our results in this section provide a few instances where a visualization based on our sampling methods can lead to a better understanding of the network. They are not comprehensive; they rather show how a combination of our sampling schemes and growth processes can provide a tool for analyzing and mining large networks. More results can be found online at www.cs.ualberta.ca/∼database/ALVIN. A demo of ALVIN is also presented in the World Wide Web Conference [30].

6 Related Work

It has been noted that *layout*, *abstraction*, *focus* and *interaction* form the basis of visualizing large networks [23]. Our work addresses the issues of focus, interaction and partly abstraction for large graphs through the use of sampling; any general graph layout algorithm can be used with our scheme.

There has been past work on layout and encoding schemes that trade off the generality for scalability and clarity and can scale-up to large trees or more specific graphs. In particular, Munzner [24] constructs spanning trees to represent the structure of a class of graphs with more tree-like structures, referred to as quasi-hierarchical graphs. The resulting tree is drawn inside a ball with fisheye distortion used to provide a focus-context view. Abello et al. [1] propose a hierarchical partitioning of the nodes based on characteristics such as the geographical locations that the nodes may represent. Using these partitions, different navigation and visualization schemes can be constructed [2]. Our work is different from these in that we don't make any assumption on the structure of the network or the characteristics of the nodes. It shouldn't be hard to integrate our work with these methods to allow the visualization to scale up to either larger or more general networks.

The work on general multiscale abstraction methods allows one to visualize either the global structure or the smaller components of a large network (e.g. [4, 18]). These methods usually do a clustering of the network and provide a coarser visualization between the clusters and a finer visualization within each cluster but not both at the same time. Gansner et al. [14] propose a notion of a hybrid graph which allows the region of interest to be viewed in a finer level and within the coarser graph. Other abstraction techniques include, but are not limited to, the work of Noik [25], Plaisant et al. [29] and Herman et al. [17]. Our work is orthogonal and compliments these abstraction methods. Our methods can be applied to a coarser view of a network when the coarser view is still too large to be fully visualized. Our use of biased sampling for focusing makes our work different from standard fisheye focusing techniques [13]. Our sampling strategies can also be applied as a pre- or post-processing step, allowing the integration of our method with other abstraction and focusing techniques.

Related to sampling from a large database, a number of algorithms have been proposed for efficiently sampling from a single table and also from the results of set union, intersection and join [26, 7]. A survey of these techniques before 1994 is given by Olken [27]. Sampling is now supported in major commercial databases and is also part of the recent SQL standard [15].

Related to our work is also the more general work on analyzing social networks (e.g. [33], [3]), mining graphs [28], URL sampling [16, 5] and analyzing the graph structure of the Web [6].

7 Conclusions

A new probabilistic approach for effectively searching and visualizing large networks is proposed, where only a sample instead of the entire network is visualized. There is no concept of a unique visualization in this scheme; instead there are many possible visualizations, each corresponding to some random sample of the network. The effectiveness of a sample and, as a result, a visualization that is based on that sample depends on the presence of the desirable patterns of the parent network in the sample. We have provided some evidence to show that indeed such patterns are preserved in a sample. Given the limitations of the screen and the size of a sample, our proposed scheme allows the search to be localized, thus increasing the ratio of sample size to the size of the desired network and removing possible biases due to the sample size.

Our work touches some of the problems related to visualizing a sample of a network. There are a number of issues that are open to further research. First, sampling has been largely used to approximately answer aggregation queries on large data sets, but there is not much work on finding sampling strategies that can preserve either the local or global properties of a network. Further studies on the subject can lead to more effective visualization schemes. Second, our work treats visualization as an incremental process that may lead to the goal after a number of growths. After each growth, a layout algorithm must be invoked to properly place the network on the canvas. A new layout may not be coherent with the old one and the elements in both layouts can be placed in different locations of the screen. Further research may look into algorithms that can preserve the locality of the nodes and still generate an effective layout after each growth.

Acknowledgments

This work is supported by Natural Sciences and Engineering Research Council of Canada. We like to thank the anonymous reviewers for their comments.

References

[1] J. Abello, I. Finocchi, and J. Korn. Graph sketches. In *Proc. of the IEEE Symposium on Information Visualization*, pages 67–70, San Diego, October 2001.

[2] J. Abello, J. Korn, and M. Kreuseler. Navigating giga-graphs. In *Proc. of the working conference on advanced visual interfaces (AVI)*, 2002.

[3] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[4] D. Auber, Y. Chiricota, F. Jourdan, and G. Melancon. Multiscale visualization of small world networks. In *Proc. of IEEE Symposium on Information Visualization*, pages 75–81, 2003.

[5] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz. Approximating aggregate queries about Web pages via random walks.

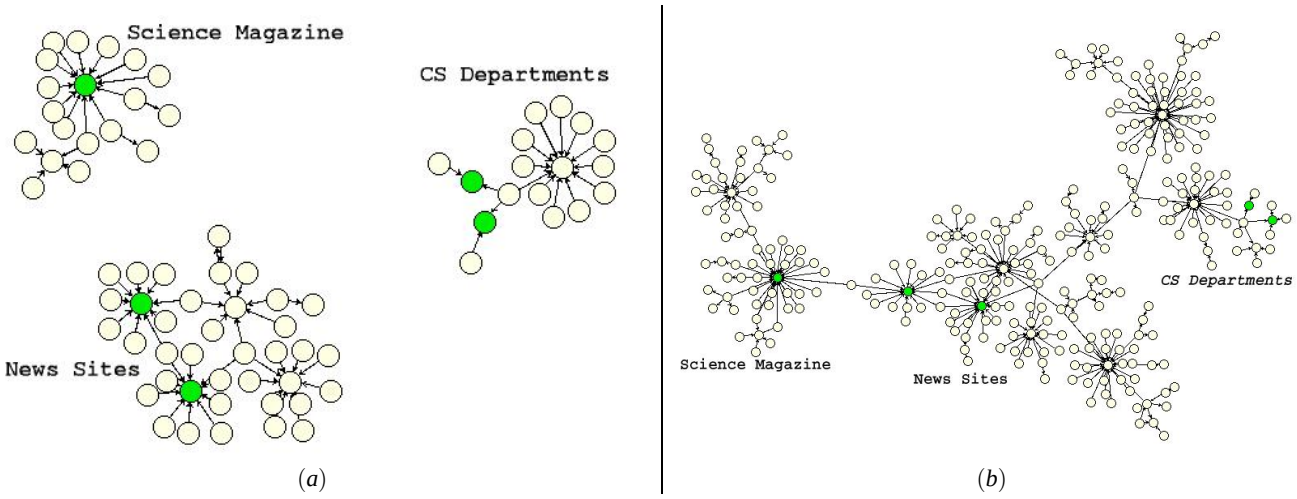

Figure 11: *A global growth (a) with the sample size set to 0.1% and (b) with the sample size set to 0.2%. In both cases, the seed set colored in green.*

In *Proc. of the VLDB Conference*, pages 535–544, Cairo, September 2000. Morgan Kaufmann.

[6] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. In *Proc. of the World Wide Web Conference*, pages 309–320, Amsterdam, May 2000.

[7] S. Chaudhuri, R. Motwani, and V. Narasayya. On random sampling over joins. In *Proc. of the SIGMOD Conference*, pages 263–274. ACM Press, 1999.

[8] A. Crosby. *The Measure of Reality: Quantification in Western Europe, 1250-1600*. Cambridge University Press, 1997. A summary is online at www.stolaf.edu/other/ql/crosby.html.

[9] S. Dill, R. Kumar, K. Mccurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins. Self-similarity in the Web. In *Proc. of the VLDB Conference*, pages 69–78, September 2001.

[10] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[11] O. Frank. Estimation of the number of connected components in a graph by using a sampled subgraph. *Scandinavian Journal of Statistics*, 5:177–188, 1978.

[12] A. Frick, A. Ludwig, and H. Mehldau. A Fast Adaptive Layout Algorithm for Undirected Graphs. In *Proceedings of Graph Drawing '94*, *Lecture Notes in Computer Science 894*, pages 388–403. Springer-Verlag, 1994.

[13] G. Furnas. Generalized fisheye views. In *Proc. of the Conference on Human Factors in Computing Systems*, pages 16–23. ACM, 1986.

[14] E. Gansner, Y. Koren, and S. North. Topological fisheye views for visualizing large graphs. In *Proc. of the IEEE Symposium on Information Visualization*, 2004.

[15] P. Haas and C. Koenig. A bi-level bernoulli scheme for database sampling. In *Proc. of the SIGMOD Conference*, pages 275–286, Paris, 2004. ACM Press.

[16] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform url sampling. In *Proc. of the World Wide Web Conference*, pages 295–308, Amsterdam, May 2000. Elsevier Science.

[17] I. Herman, M. Marshall, G. Melancon, D. Duke, M. Delest, and J.-P. Domenger. Skeletal images as visual cues in graph visualization. In *Proc. of the Data Visualization*, pages 13–29, 1999.

[18] J. Huotari, K. Lyytinen, and M. Niemel. Improving graphical information system model use with elision and connecting lines. *ACM Transactions on Computer-Human Interaction*, 11(1):26–58, March 2004.

[19] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[20] A. Klovdahl. A note on images of networks. *Social Networks*, 3:197–214, 1981.

[21] V. Krishnamurthy, J. Sun, M. Faloutsos, and S. Tauro. Sampling Internet topologies: how small can we go? In *Proc. of International Conference on Internet Computing*, Las Vegas, 2003.

[22] LEDA. class library. *www.algorithmic-solutions.com/enleda.htm*.

[23] A. Mendelzon. Visualizing the world wide web. In *Proc. of the working conference on advanced visual interfaces (AVI)*, 1996.

[24] T. Munzner. *Interactive visualization of large graphs and networks*. PhD thesis, Stanford University, 2000.

[25] E. Noik. *Dynamic fisheye views: combining dynamic queries and mapping with database views*. PhD thesis, University of Toronto, 1996.

[26] F. Olken. *Random Sampling from Databases*. PhD thesis, University of California at Berkeley, 1993.

[27] F. Olken and D. Rotem. Random sampling from databases - a survey. *Statistics and Computing*, 5(1):25–42, March 1995.

[28] C. Palmer, P. Gibbons, and C. Faloutsos. Data mining on large graphs. In *Proc. ACM Intl. Conf. on SIGKDD*, pages 81–90, 2002.

[29] C. Plaisant, J. Grosjean, and B. Bederson. Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proc. of the IEEE Symposium on Information Visualization*, pages 57–66, Boston, October 2002.

[30] D. Rafiei and S. Curial. ALVIN: a system for visualizing large networks. In *Poster Proc. of the World Wide Web Conference*, May 2005.

[31] S. Thompson. *Sampling*. Wiley, 2nd edition, 2002.

[32] Tulip. *www.tulip-software.org*.

[33] D. Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton University Press, 1999.

A A RANDOM ORDERING OF THE EDGES IN THE RUNNING EXAMPLE OF SECTION 4

Our examples in Section 3 use the following random ordering of the edges shown in Figure 8: (b1,b10), (a1,a5), (c1,c3), (b1,b9), (d1,d2), (d3,d4), (a7,b11), (c2,c6), (c11,c7), (c11,c6), (a1,a7), (a5,a13), (c3,c8), (b1,b7), (a5,a14), (c3,c7), (a5,a12), (a7,a16), (a2,a9), (c1,c4), (c2,c9), (c4,c9), (d4,d1), (a2,a10), (c2,c8), (b1,b4), (c1,c5), (a1,a4), (b1,b3), (a4,a11), (b1,b5), (c3,c6), (a1,a2), (d2,d3), (c1,c2), (b1,b6), (c2,c7), (c11,c9), (c5,c9), (a1,a3), (b1,b11), (a7,a15), (c11,c10), (a1,a6), (b1,b8), (c4,c6), (e2,e3), (e1,e2), (c4,c8), (b1,b2), (c11,c8), (c5,c10), (a2,a8), (c4,c7), (c3,c9).