Efficient Identification of Web Communities

Gary William Flake
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
flake@research.nj.nec.com

Steve Lawrence
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
lawrence@research.nj.nec.com

C. Lee Giles
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
giles@research.nj.nec.com

ABSTRACT

We define a community on the web as a set of sites that have more links (in either direction) to members of the community than to non-members. Members of such a community can be efficiently identified in a maximum flow / minimum cut framework, where the source is composed of known members, and the sink consists of well-known non-members. A focused crawler that crawls to a fixed depth can approximate community membership by augmenting the graph induced by the crawl with links to a virtual sink node. The effectiveness of the approximation algorithm is demonstrated with several crawl results that identify hubs, authorities, web rings, and other link topologies that are useful but not easily categorized. Applications of our approach include focused crawlers and search engines, automatic population of portal categories, and improved filtering.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering, information filtering*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*

1. INTRODUCTION

The rapid growth of the World Wide Web has made more information freely available than ever before. While this growth has ushered a boon to most levels of society, it has also created a dilemma for search engine designers that have to balance a number of conflicting goals in order to make search engines practical in the real-world.

One conflict hinges on the shear number of indexable web pages (now over 10^9 [1]). Ideally, search engine crawlers could sample the indexable web often enough to insure that results are valid, and broadly enough to insure that all valuable documents are indexed. However, the most recent studies have shown that no search engine covers more than about 16% of the web, and the union of 11 major search engines covers less than 50% of the web [2]. Moreover, search engines are often out-of-date partially due to limited crawling speeds and the low average life-span of web pages.

A second dilemma for search engines resides in the balance between precision and recall of query results. Since most search engines rank results with a topical measure, broad queries to general search engines can easily return thousands of results, thus yielding high recall at the expense of precision. Web portals such as Yahoo! approach the problem from the other extreme by organizing a very small subset of the web into a hierarchical structure which can yield high precision for a search but with low recall.

Motivated by these two problems, we introduce a definition of a web *community* that may enable web crawlers to effectively focus on narrow but topically related subsets of the web and also enable search engines and portals to increase the precision and recall of search results. We define a community to be a set of web pages that link (in either direction) to more web pages in the community than to pages outside of the community. Superficially, this definition appears problematic because it depends on a community being identified before an individual web page can be tested for membership. In fact, in the absence of any *a priori* information, our definition of a community is NP-complete because it maps into a family of graph partitioning problems. However, by exploiting various properties of the web (e.g., that so called "hubs" and "authorities" can be easily identified) identifying a web community becomes identical to solving the s-t maximum flow network problem, which has many efficient polynomial time solutions.

The remainder of this paper is divided into four additional sections and an appendix. In Section 2 we give an overview of previous work on classifying hyper-linked documents by link analysis, discuss graph cut methods, and provide an introduction to network flow analysis. The purpose of this background section is to illustrate how our proposed methods complement previous work and exploit regularities of the web to yield fast methods for identifying web communities.

Section 3 contains the bulk of our analysis, where we prove that identifying a community is identical to solving the s-t maximum flow problem. Since our formulation of web communities exploits regularities of the web, we also prove some requirements and limitations of the algorithm. After han-

Appears in the Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000), pages 150–160, August 20–23, 2000, Boston, MA, USA.

dling the idealized case (where the entire web is considered the input to the flow algorithm) we introduce an approximation to the idealized method that operates over a graph induced by a crawl of fixed depth. The approximation algorithm is combined with an expectation-maximization procedure that iterates crawls by bootstrapping previous results.

In Section 4, we demonstrate the effectiveness of our algorithms by showing the results from three focused crawls directed by the procedures described in Section 3. The first community that we identify is topically related around a research area; the second community is focused around a set of institutions; and the third community was found by starting with a single individual's home page.

Finally, Section 5 contains our conclusions and discusses future areas of research for the methods that we introduce. Additionally, we present a new maximum flow algorithm in the appendix that may offer scalable solutions on graphs the size of the entire web.

2. BACKGROUND

A considerable amount of research has focused on analyzing collections of hyper-linked documents and structures. These works have been very cross-disciplinary, occurring in hyper-media, WWW, sociology, bibliometric, and software engineering circles. We summarize a small subset of these fields to give our own work the proper context.

2.1 Link Analysis

One of the earliest uses of link structure is found in the analysis of social networks [3], where network properties such as cliques, centroids, and diameters are used to analyze the collective properties of interacting agents. The fields of citation analysis [4] and bibliometrics [5] also use the citation links between works of literature to identify patterns in collections. Co-citation [6] and bibliographic coupling [7] are two of the more fundamental measures used to characterize the similarity between two documents. The first measures the number of citations in common between two documents, while the second measures the number of documents that cite both of two documents under consideration. Methods from bibliometrics have also been applied to the problem of mining web documents [8].

Bibliometrics techniques can be thought of as local in nature because they typically consider only the local link properties between two documents. Of course, similarity metrics such as co-citation and bibliographic coupling can be used along with classical clustering techniques, such as k-means [9], to reduce the dimensionality of the document space, thus identifying documents in a community that is centered about a cluster centroid. More radical forms of dimensionality reduction have used this basic idea to cluster literature databases with over 150 thousand documents [10]. However, applying these methods to systems such as the web, with over 10^9 documents, would obviously be challenging.

Recently, Kleinberg [11] showed that the HITS algorithm, which is strongly related to spectral graph partitioning and methods used by the Google search engine [12], can identify "hub" and "authority" web pages. A hub site links to many authority sites and an authority site is linked to by many hubs. Thus, the definition of the two page types is recursive and mutually reinforcing. HITS is especially effective when one desires results to be ranked by a measure of importance that reflects some notion of usefulness and relevance as determined by link topology.

More formally, if A is the adjacency matrix of a directed graph, HITS finds the left and right handed eigenvectors of $A^T A$ with the largest eigenvalue. By definition, all components of the dominant eigenvector will have positive components. The web pages that correspond to the largest components of the right-handed (or left-handed) eigenvector are the authorities (or hubs).

We consider HITS to be complementary to our own work, as our algorithm requires "seed" web sites to be used as the starting point for a focused crawl. Hubs and authorities are very useful for identifying key sites related to some community and, hence, should work well as seeds to our method. HITS has also been used [13] to extract related[1] documents; however, using HITS for enumerating all members of a community may be problematic because the communities that one is interested in may be overshadowed by a more dominant community. To solve this problem with HITS, one must extract multiple eigenvectors in order to isolate the smaller community that one is interested in.

Another issue for HITS is that it may have limited use in identifying communities that form web rings and small-world networks without dominating members. In the case of simple web rings, the adjacency matrix of the subgraph of the community forms a permutation matrix. In this case, the matrix $A^T A$ equals the identity matrix and will, therefore, have a flat spectrum of eigenvalues. Our method is not troubled by such link topologies.

2.2 Graph Cuts and Partitions

If time and space complexity issues were irrelevant, then one could identify tightly coupled communities by solving the problem as a balanced minimum cut problem, where the goal is to partition a graph such that the edge weight between the partitions is minimized while maintaining partitions of a minimal size. In this framework, communities obey our defining characteristic of having more edges inside the community than outside. Unfortunately, the most generic versions of balanced minimum-cut graph partitioning are NP-complete [14]. On the other hand, if the constraint on the partition sizes is removed, then the problem lends itself to many polynomial time algorithms [15]; however, under this formulation, solutions will often be trivial cuts that leave one partition very small relative to the size of the original graph.

Intuitively, balanced minimal cuts are hard because of the vast number of balanced partitions that one can place on a vertex set. Unrestricted minimal cuts are easier because there are relatively few trivial (highly unbalanced) partitions in a graph. In any event, neither approach lends itself to identifying communities as we have defined them.

[1]Gibson *et al.* [13] also use the term *community* to describe their discovered collections; however, their definition is not strictly consistent with our own, so we avoid using it here.

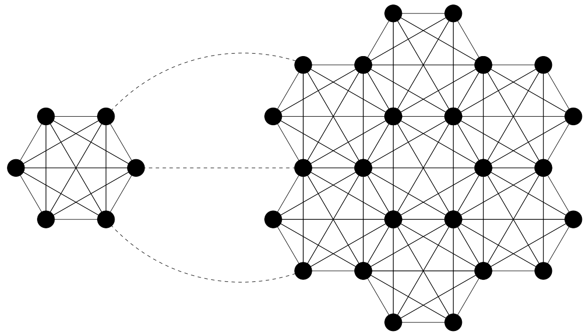

Figure 1: Maximum flow methods will separate the two subgraphs with any choice of s and t that has s on the left subgraph and t on the right subgraph, removing the three dashed links.

2.3 Maximum Flow and Minimal Cuts

The s-t maximum flow problem is defined as follows. Given a directed graph $G = (V, E)$, with edge capacities $c(u, v) \in \mathbb{Z}^+$, and two vertices, $s, t \in V$, find the maximum flow that can be routed from the source, s, to the sink, t, that obeys all capacity constraints. Intuitively, if edges are water pipes and vertices are pipe junctions, then the maximum flow problem tells you how much water you can move from one point to another.

The famous "max flow-min cut" theorem of Ford and Fulkerson [16, 17] proves that the maximum flow of the network is identical to the minimum cut that separates s and t. Many polynomial time algorithms exist for solving the s-t maximum flow problem, and applications of the problem include VLSI design, routing, scheduling, image segmentation, and network reliability testing [18].

The maximum flow problem is well-suited to the application of identifying web communities because, unlike the balanced and unbalanced graph partition problems, it is computationally tractable and it allows us to exploit *a priori* knowledge about the underlying graph.

Most modern solutions to the maximum flow problem operate under the assumption that the entire graph under consideration can be examined easily. This is obviously not the case with the web, as the entire graph that corresponds to the web is vastly larger than any single computer can store in main memory. Nevertheless, one of the simplest maximum flow algorithms—the shortest augmentation path algorithm [19]—can solve the problem by examining only the portions of the graph that arise when locating shortest paths between the source and sink nodes. Thus, it should be possible to solve a maximum flow problem on the entire web with today's computers.

3. WEB COMMUNITIES

In this section we formalize our definition of a web community and describe methods to identify them. First, we handle the ideal case, when the entire web can be used for the calculation. Afterwards, we explore approximate methods and describe the method used in our experiments.

3.1 Ideal Communities

We first define communities in terms of undirected graphs where each edge has unit capacity. Thus, the graph induced from the web would have edge directions removed.

DEFINITION 1. *A* COMMUNITY *is a vertex subset $C \subset V$, such that for all vertices $v \in C$, v has at least as many edges connecting to vertices in C as it does to vertices in $(V - C)$.*

Note that this definition is slightly recursive in that it leads to statements of the form "a Pokémon web site is a site that links to or is linked by more Pokémon sites than non-Pokémon sites." Figure 1 shows an example of a community (on the left) being separate from the rest of the graph (on the right). We now define two quantities that characterize the adequacy of the source and sink.

DEFINITION 2. *Let the* SOURCE LINK COUNT, *$s^\#$, refers to the number of edges between s and all vertices in $(C - s)$. Similarly, let the* SINK LINK COUNT, *$t^\#$, refers to the number of edges between t and vertices $(V - C - t)$.*

THEOREM 1. *A community, C, can be identified by calculating the s-t minimum cut of G with s and t being used as the source and sink, respectively, provided that both $s^\#$ and $t^\#$ exceed the cut set size. After the cut, vertices that are reachable from s are in the community.*

PROOF. *Assume by way of contradiction that the s-t minimum cut solution resulted in a cut that had some vertex, $v \notin C$, being reachable from s. Since $v \notin C$, we know that v connects to more vertices in $(V - C)$ then C. However, in this case moving v to the other side of the cut would result in a more efficient cut, which violates the initial assumption that this is a minimum cut and that a vertex such as v could exist. A similar argument can be made for the case of $v \in C$ being reachable from t.* □

The conditions placed on $s^\#$ and $t^\#$ in Theorem 1 are to disallow the trivial solution of excising the source or sink set from the rest of the graph. Since it is possible to generalize the definition of a community so that weighted edges are permitted, using higher weights near the source vertex is another way to discourage trivial cuts. We discuss this idea in more detail in the next subsection.

A community that consists of hundreds or thousands of web pages will obviously have many links outside of the community, especially when one considers the impact of bookmark web pages which often have many links to unrelated sites. As a result, the condition on $s^\#$ is problematic if only a single source vertex is used. We can address this problem by using multiple *seeds* connected from a single virtual source node. The trick works by assigning infinite capacities to the edges that connect the virtual source to the seeds.

The choice for a sink vertex should be easy to automate because of some well-known properties of the web. Since the web is approximately a small-world network, with a power

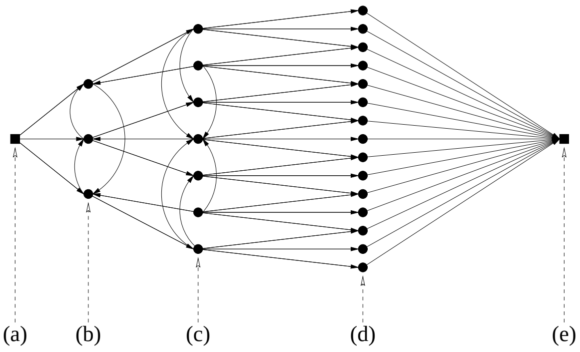

Figure 2: Focused community crawling and the graph induced: (a) The virtual source vertex; (b) vertices of seed web sites; (c) vertices of web sites one link away from any seed site; (d) references to sites not in (b) or (c); and (e) the virtual sink vertex.

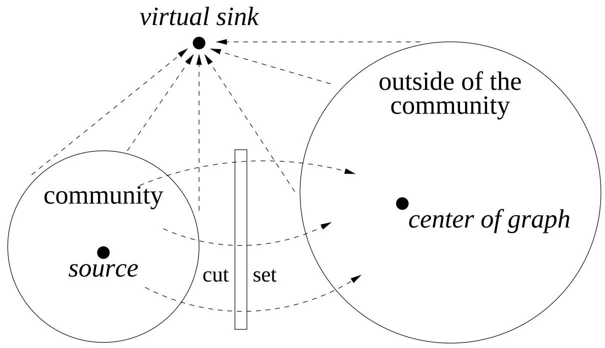

Figure 3: Locating a cut even when a good sink is not available: All edge capacities in the graph are multiplied by a constant factor, k. Unit capacity edges are added from every vertex to a new virtual sink.

law distribution on the inbound and outbound links [20], web portal sites such as Yahoo! should be very close to the center of the web graph. Thus, by using the top-levels of a small collection of web portals as a virtual sink, the maximum flow formulation of communities should be able to extract a closely knit community from the rest of the web, because the center of the graph is very general in the sense that there is a short path to any other site on the web.

3.2 Approximate Communities

One problem with calculating ideal communities is that it requires rapid access to the inbound and outbound links for many web sites. In principle, one could use a resource such as the Internet Archive to maintain a snapshot of the web graph on a single computer. We intend to do this in future research; however, we have developed an approximate method for identifying web communities that appears to work well in practice.

Figure 2 illustrates how our focused crawler retrieves pages and the graph that is induced by the crawl. (For another example of focused crawling, see [21].) The crawl begins with the seed web pages, shown as set (b) in the figure, and finds all pages that link to or from the seed set. Outbound links are trivially found by examining the HTML of the page. Inbound links are found by querying a search engine that supports the `link:` modifier (such as Google or AltaVista).

Once the URLs from set (c) are identified, their HTML is downloaded and all outbound links are recorded. Some of these outbound links may refer to pages already crawled (e.g., links from (c) to (c) and (c) to (b)); however, many of the outbound links from (c) will refer to pages not downloaded (from set (d)). The pages corresponding to set (d) are effectively treated as a composite sink vertex, as each is linked to a virtual sink vertex.

In Figure 2, all edges between sets that start and end in either of sets (b) or (c) are made bidirectional. Thus, we treat inbound and outbound links identically when they are close to the seed set. All other edge directions (from (a) to (b), (c) to (d), and (d) to (e)) are preserved.

Notice that in our approximation we are not using any true web page as a sink, but are instead using an artificial vertex that is connected to by the vertices in our graph that are most distant from the source. We justify this procedure with the following observations.

Suppose we knew of a source vertex with sufficient link degree to separate a community, and that, when using the true center of the graph, flow techniques produce the cut as shown in Figure 3. We would like to be able to approximate this cut even when we do not know the true identity of the graph center. If we multiply all edge capacities in the graph by a constant factor k, and add a virtual sink to the graph such that all vertices are connected with a unit capacity edge to the virtual sink, then under what conditions will flow methods with the original source and the virtual sink yield the same partitioning?

THEOREM 2. *Let c denote the total capacity of the original cut set and let N_s and N_t denote the number of vertices in the community, C, and not in the community, $(V-C)$. If the condition $1 < k < N_t/c$ holds, then the augmented graph as shown in Figure 3 will produce the same partitioning as the un-augmented graph, except that all edges from C to the virtual sink, $\hat{t}$, will be cut as well.*

PROOF. *We prove this constructively in four steps. First, multiplying all edge capacities by k will obviously not change the solution because the flow relationship between all vertices remains unchanged except for constant scalings. Second, if we connect all vertices in $(V-C)$ to the virtual sink, $\hat{t}$, then the bottleneck will remain at the original cut set provided that $N_t > ck$, which is one of our assumptions. Third, for all vertices in $v \in C$, connect v to $\hat{t}$ with a unit capacity edge. Since $k > 1$, cutting this newly added edge is more efficient than removing v from the community. Fourth, we require that the trivial cut of removing all edges to $\hat{t}$ be more expensive than performing the original cut along with removing all edges from C to $\hat{t}$. This final condition is true if*

$$N_s + ck < N_s + N_t,$$

which is implied by our assumption $k < N_t/c$. $\square$

```
procedure Focused-Crawl(graph: G = (V, E);
                        vertex: s, t ∈ V)
   while number of iterations is less than desired do
      Set k equal to the number of vertices in seed set.
      Perform maximum flow analysis of G,
         yielding community, C.
      Identify non-seed vertex, v* ∈ C,
         with the highest in-degree relative to G.
      for all v ∈ C such that in-degree of v equals v*
         Add v to seed set.
         Add edge (s, v) to E with infinite capacity.
      end for
      Identify non-seed vertex, u*,
         with the highest out-degree relative to G.
      for all u ∈ C such that out-degree of u equals u*
         Add u to seed set.
         Add edge (s, u) to E with infinite capacity.
      end for
      Re-crawl so that G uses all seeds.
      Let G reflect new information from the crawl.
   end while
end procedure
```

Table 1: The complete focused crawl procedure.

Of course, our method of wiring every vertex in set (d) of Figure 2 is actually more extreme than wiring every vertex to the virtual sink. We heuristically make this choice because we have anecdotally observed that many members of a web community are often one link away from our seed sets and that most web pages two links away do not belong in the community due to the rapid branching factor of the web.

Intuitively, our method of using a virtual sink is very similar to methods used in image segmentation [22]. In the maximum flow formulation of image segmentation, adjacent pixels are connected with edge capacities as a function of gray level similarity. All pixels on the perimeter of an image are connected to a virtual sink. Thus, segmenting an image in this manner is similar to finding a "community" of pixels that are more similar to each other than pixels outside of the segment.

One other deviation that we make in our implementation is that we only use the constant scaling factor k on edges that extend from set (b) to set (c) of Figure 2. We do this because of the same anecdotal observations that larger capacities should be used closer to the seed vertices.

3.3 Expectation Maximization

As describe in the previous subsection, our method for identifying web communities has only limited success when a small number of seed web pages are provided (say, less than four). The problem is that with a small number of seeds, only a relatively small subset of a community can be identified because the combined out degree of the seeds has to be larger than the cut size (the number of edges removed). Thus, we need to augment our procedure with a method for identifying new seeds.

We solve this problem with a method inspired by the Expectation Maximization (EM) algorithm [23]. The EM algorithm is a two-step process that iteratively applies estimation (the "E" step) and maximization (the "M" step). In our case, the "E" step corresponds to using the maximum flow algorithm to identify a subset of the community. The strongest newly discovered web site (in terms of link degree), are relabelled as seeds. We then partially re-crawl from the new seeds to induce a new graph. The maximum flow procedure is rerun, and the process iterates.

Our online procedure for identifying web communities is described in pseudo-code in Table 1. The procedure loops for a fixed number of iterations, sets k from the previous section equal to the cardinality of the seed set, performs maximum flow analysis, and uses an EM-inspired method for relabelling newly discovered web pages as seeds.

We typically run the procedure for four iterations. We have noticed that with a larger number of iterations the algorithm reaches a fixed-point; however, we can not analytically support that the procedure will always terminate with a fixed-point, nor can we prove the deviation of the approximate method from the ideal at this time. Nevertheless, in the next section we summarize our experimental results, which offer support to our claim that the maximum-flow based focused crawler can identify interesting web communities.

4. EXPERIMENTAL RESULTS

To test our maximum flow method of identifying communities, we used the focused crawler described in the previous section to extract web pages related to three different seed sets. The first community is topically centered around the research area of support vector machines (SVM); the second community is centered around the Internet Archive; and the third community is centered around Ronald Rivest.

4.1 Support Vector Machine Community

We selected the support vector machine community as an interesting test case because this research area began around five years ago. As a result, the community is relatively small compared to other research communities, has a fair number of prominent researchers, and is not listed in any portal that we know of.

For this crawl, we completely ignored links where the source and destination resided in the same domain. Filtering links in this manner makes the procedure more robust against "inbreeding," i.e., web sites with an excessive amount of internal linking. However, the internal pages of a web site can still be pulled into a community by our crawler provided that some other site links to the first site's internal page.

Our seed set consisted of four URLs:

```
http://svm.first.gmd.de/
http://svm.research.bell-labs.com/
http://www.clrc.rhbnc.ac.uk/research/SVM/
http://www.support-vector.net/
```

All of the above are the front pages of groups that have a fair number of people working on SVM research. We iterated our EM procedure four times. In the final iteration, over 11,000 URLs had vertices in the graph induced from the crawl. The discovered community contained 252 member web pages in it. We assigned a score to each page equal to the sum of the number of inbound and outbound links from a page that connects to other community members.

URL	Description
http://www.research.att.com/info/vlad	Vladimir Vapnik's home page (inventor of SVMs)
http://svm.research.bell-labs.com/	Lucent Technologies SVM page (seed)
http://www-ai.cs.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html	page for SVM light, a popular software package
http://vision.ai.uiuc.edu/mhyang/svm.html	a hub site of SVM links
http://svm.first.gmd.de/	GMD First SVM page (seed)
http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html	copy of SVM light page
http://www.research.att.com/~lewis/reuters21578.html	text categorization corpus
http://www.clopinet.com/isabelle/Projects/SVM/applist.html	SVM application list
http://www.research.microsoft.com/~jplatt/svm.html	John Platt's SVM page
http://www.csi.uottawa.ca/~marchand/publications/Research.html	Research interests of Mario Marchand
http://www.ai.mit.edu/projects/cbcl/acai99/acai99.html	SVM workshop page
http://svm.first.gmd.de/publications.html	GMD First SVM publication list
http://svm.first.gmd.de/nips97/book.html	Book: Advances in Kernel Methods - SVM Learning
http://www.mpik-tueb.mpg.de/people/personal/bs/svm.html	Bernhard Schölkopf's SVM page
http://svm.first.gmd.de/people.html	GMD First hub page of SVM researchers
http://www.dcs.qmw.ac.uk/~yongmin/links.html	Yongmin Li's links to SVM pages
http://www-ai.cs.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light_v1.00.eng.html	and yet another SVM light page
http://www-ai.cs.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light_v2.01.eng.html	amazingly, another SVM light page
http://svm.first.gmd.de/nips97/abstracts.html	NIPS SVM workshop abstract page
http://svm.first.gmd.de/links.html	GMD First SVM links
http://wwwsyseng.anu.edu.au/lsg/	Learning System Group of ANU
http://svm.first.gmd.de/nips98/workshop.html	NIPS*98 workshop on large margin classifiers
http://math.la.asu.edu/~kawski/seminars/fall97.html	control theory seminar (with links to SVM material)
http://www.isis.ecs.soton.ac.uk/resources/svminfo/	ISIS SVM page
http://www.cogs.susx.ac.uk/users/jonh/r_index2.html	Jonathan Howell's home page
http://www-ai.cs.uni-dortmund.de/thorsten/svm_light.html	another SVM light page
http://svm.first.gmd.de/nips97/workshop.html	NIPS*97 Workshop on Support Vector Machines
http://research.microsoft.com/~jplatt/svm.html	alternate URL for Platt's home page
http://www.rpi.edu/~demira/	Ayhan Demiriz, SVM researcher
http://www.lans.ece.utexas.edu/course/ee380l/1999fall/index.shtml	data mining page
http://akpublic.research.att.com/info/vlad	mirror of Vladimir Vapnik's page
http://www.isis.ecs.soton.ac.uk/research/svm/	another ISIS SVM page
http://www.cse.ucsc.edu/research/compbio/genex/svm.html	gene expression analysis with SVMs
http://www.cc.gatech.edu/classes/cs4803c_98_fall/sv.html	SVM links page for course
http://www.first.gmd.de/~smola/	Alex Smola's old home page, SVM researcher
http://lara.enm.bris.ac.uk/~cig/links.htm	U. of Bristol machine learning links
http://www.ai.eecs.uic.edu/~bkao/ml.html	Machine Learning & Data Mining Hotlist
http://www.support-vector.net/	support vector machines, the book (seed)
http://www.research.microsoft.com/~jplatt/smo.html	John Platt's SVM SMO page
http://www.ai.mit.edu/courses/6.893/related-info.html	machine learning and neural network links

Table 2: Members of the SVM community with the highest scores, sorted in descending order.

Table 2 contains the first forty web pages as ranked by our naive scoring method. Interestingly, Vladimir Vapnik's home page (whom may be considered the founder of the SVM community) is at the top of the list. As can be seen, almost all of the returned results are strongly related to SVM research. Arguably, four of the first forty can be considered false positives because they are more strongly related to data mining, neural networks, or object recognition methods. However, all of these pages have at least some relationship to the SVM community.

The community members with the lowest score are arguably more interesting. All ten lowest scoring web pages are home pages of researchers with publications in SVMs.[2] The rest of the web community contains a large number of researchers and their students, SVM software, commercial links to SVM books, and SVM conference and workshop announcements.

To be sure, the approximated community also contains a few false positives, which are usually neural network or data mining pages. However, the most flagrant false positive was a link to a script archive.

4.2 The Internet Archive Community

We chose the Internet Archive[3] (IA) to serve as the second example because it is an institute that straddles many different real-world communities: information retrieval, library science, archiving, visualization, etc. Because the IA is so broad, we seeded this crawl with eleven URLs found on the IA's resources page and the IA's home page as well:

```
http://www.archive.org/
http://www.clir.org/diglib/dlfhomepage.htm
http://www.dlib.org/
http://webreference.com/internet/statistics.html
http://trec.nist.gov/
http://www.intermemory.org/
http://www.informedia.cs.cmu.edu/
http://www.digitalmuseums.org/index.html
http://www.cybergeography.com/atlas/surf.html
http://www.peacockmaps.com/
http://www.cs.bell-labs.com/~ches/map/gallery/
http://www.archivists.org/
```

As with the previous crawl, we excluded all links internal to a domain from consideration. We iterated the EM algorithm twice, which produced over 7000 URLs. The final iteration of the maximum flow algorithm trimmed the resulting graph down to a community of 289 web pages.

The top forty results are shown in Table 3. As can be seen, there is a broad mix of media surveying companies, archival organizations, library science, Internet statistics, digital library institutions, and other web sites closely related to the mission of the IA.

The remainder of the community contains news stories about the IA, information retrieval conference announcements, researchers' home pages, plus others. The three sites tied for the lowest score were:

```
http://www.informedia.cs.cmu.edu/
http://www.imls.fed.us/librlinks.htm
http://www.digitalmuseums.org/index.html
```

All three of the lowest scoring URLs are still related to the IA. The most obvious false positive that we found was `http://mail.yahoo.ca/` which slipped into the community because six Yahoo! pages related to archiving and digital libraries made it into the community as well. Since the crawl only extends to a depth of two from the seeds, and since only two applications of the EM algorithm were used, we suspect that had the inbound links to the offending URL been loaded, it would have been possible for the flow algorithm to remove the false positive.

4.3 The "Ronald Rivest" Community

For our final example, we decided to identify a community around a specific individual. We chose Ronald Rivest as a candidate because he is well-known and because there are a large number of web pages that reference his work on encryption and his book "Introduction to Algorithms" [17]. Our single seed URL was `http://theory.lcs.mit.edu/~rivest`.

Because we used a single seed, we found it necessary to make the first iteration of the crawl use internal links as well as external links. A total of four EM iterations were used, and the final three used only external links.

After the final iteration, over thirty-eighth thousand URLs had vertices in the induced graph. The maximum flow algorithm trimmed this down to 150 pages.

Once again, the top forty results are shown in Table 4. The most striking result is that Rivest's co-authors on "Introduction to Algorithms" (Thomas H. Cormen and Charles E. Leiserson) appear as the first and third results. The book's web page was ranked in the 92nd position.

As can be seen, the other web pages in the top forty are all related to cryptography, security, and MIT. Twenty-three URLs tied for the lowest score. Of these, all were personally relevant to Rivest or his research, and 11 of these were bibliographies of his publications.

5. SUMMARY AND FUTURE WORK

We have defined a new type of web community that can be efficiently calculated in a maximum flow framework. We have also introduced a maximum flow-based web crawler that can approximate a community by directing a focused web crawler along link paths that are highly relevant. Our discovered communities are very cohesive in the sense that members of the community are more tightly coupled to each other than to non-members. The EM approach that we use incrementally improves the crawl results by re-seeding our crawler with highly relevant sites.

In all experiments, the runtime of the algorithms was vastly dominated by the network demands of retrieving web pages. The runtime for the actual flow algorithm never required more than one second of CPU time on a 500 Mhz Intel machine. Thus, if a portion of the web graph is locally stored, it is possible that our methods can yield results fast enough for online use.

[2] We choose not to list these URLs as we wish no inference on the ranking to be made with respect to a researcher's standing in the community.

[3] `http://www.archive.org/`

URL	Description
http://www.mediametrix.com/	Media Metrix
http://www.cc.gatech.edu/gvu/user_surveys/	GVU Center's WWW User Surveys
http://www.clir.org/diglib/dlfhomepage.htm	Digital Library Federation
http://www.peacockmaps.com/	Peacock Maps of the Internet (seed)
http://trec.nist.gov/	Text REtrieval Conference (TREC) Home Page (seed)
http://www.archivists.org/	The Society of American Archivists (seed)
http://www.simmons.edu/~saa/scosaa.htm	Society of Am. Archivists, Simmons Coll. Student Chapter
http://www.webreference.com/internet/statistics.html	Internet Statistics
http://webreference.com/internet/statistics.html	Internet Statistics (seed)
http://www.cybergeography.com/atlas/surf.html	An Atlas of Cyberspaces- Surf Maps (seed)
http://www.asap.unimelb.edu.au/asa/stama/stama.htm	STAMA home page, an archiving organization
http://www.fsconsult.com/presentations/	FS Consulting - Presentations & Conferences
http://www.mids.org/	Matrix Information and Directory Services
http://calafia.com/classroom/demographics.htm	Calafia Classroom: Internet Demographics
http://www.hti.umich.edu/misc/ssp/workshops/teidlf/sumrec.html	TEI and XML in Digital Libraries
http://www.teleport.com/~tbchad/stats1.html	InfoQuest! Internet Surveys & Statistics
http://www.yahoo.com/Computers_and_Internet/Internet/Statistics_and_Demographics/	Yahoo! Computers and Internet:Internet:Statistics and Demographics
http://www.fas.org/cp/netstats.htm	CyberStats Internet statistics
http://slashdot.org/articles/99/09/02/1120214.shtml	article on mapping the Internet
http://www.arma.org/	Association of Information Management Professionals
http://www.cni.org/	Coalition for Networked Information
http://www.clir.org/	Welcome to the Council on Library and Information Resources
http://dlis.dos.state.fl.us/barm/calendar.html	Bureau of Archives & Records Management calendar
http://www.searchtools.com/tools/bus.html	BUS Indexer and Search Engine
http://www.readware.com/Default.htm	Managment Information Technologies
http://www.openmarket.com/intindex/index.cfm	The Internet Index
http://www.nielsen-netratings.com/	Nielsen NetRatings
http://clrc.org/refoct99.htm	library resources calendar
http://www.ciolek.com/WWWVLPages/QltyPages/NetStudies.html	Internet Studies
http://www.dei.unipd.it/~ims/sigir99/	Evaluation of Web Document Retrieval - SIGIR'99 Workshop
http://sage.library.emory.edu/97pressrelease.html	archives at Georgia Tech and Emory
http://www.telegeography.com/	TeleGeography Home Page
http://pastime.anu.edu.au/WAR/webtrax.html	TREC-8 Web Track
http://mailer.fsu.edu/~baltman/sfa.html	The Society of Florida Archivists
http://ils.unc.edu/~geisg/dsi/dsi/	Internet2 Distributed Storage Infrastructure Initiative
http://www.focalink.com/aksystem/datapartners.html	AdKnowledge Data Partners
http://www.lib.umb.edu/newengarch/InternetResources/otherassoc.html	New England Archivists links page
http://www.thecounter.com/stats/	TheCounter.com - Global Statistics
http://www.terena.nl/news/reports/1999/report001.html	Report on the Network Storage Symposium
http://www.portugues.mct.pt/branco/white_paper.html	Computational processing of Portuguese: working memo

Table 3: Members of the Internet Archive community with the highest scores, sorted in descending order.

URL	Description
http://www.cs.dartmouth.edu/~thc/	Thomas H. Cormen's home page
http://world.std.com/~franl/crypto/rsa-guts.html	The Mathematical Guts of RSA Encryption
http://theory.lcs.mit.edu/~cel/	Charles E. Leiserson's home page
http://www.it.kth.se/~rom/cryptopeople.html	Famous people in the history of Cryptography
http://www.homeport.org/~adam/web/crypto.html	Cryptography sites
http://web.mit.edu/	Massachusetts Institute of Technology
http://www.state.in.us/digitalsignatures/links.html	general cryptography links
http://www.spektrum.de/themen/Kryptographie-text.html	Spektrum der Wissenschaft - Kryptographie
http://www.csu.edu.au/special/auugwww96/proceedings/varamu/ecom.html	Issues in Securing Electronic Commerce over the Internet
http://www.cs.unlv.edu/~larmore/Courses/CSC477/S99/	course based on "Introduction to Algorithms"
http://www.cil.lu/cil/training/books.html	Recommended Literature for Self-Study
http://www.cs.vu.nl/~aske/resume.html	Resume of Aske Plaat
http://www.heise.de/ix/artikel/1999/07/166/	German article on who's who of the WWW
http://www.daimi.aau.dk/~ulle/people.html	People Ulrik knows
http://www.uni-paderborn.de/fachbereich/AG/agmadh/vorl/InfoC/	course that uses "Introduction to Algorithms"
http://liinwww.ira.uka.de/bibliography/Misc/Intro.Algorithms.html	Bibliography on algorithms
http://boardwatch.internet.com/mag/97/jul/bwm19.html	article on encryption
http://www.mpi-sb.mpg.de/about_d.html	German computer science institute
http://www.mpi-sb.mpg.de/about.html	same as above
http://www.ainet.or.jp/~inoue/link/security.html	security links
http://mail.telstar.net/mirror/pgp/FAQ.shtml	International PGP FAQ
http://www.dca.fee.unicamp.br/pgp/FAQ.shtml	FAQ do PGP International
http://www.peachnet.edu/oiit/support/pc/	OIIT: Personal Computing
http://www.heureka.clara.net/sunrise/pgpsign.htm	Always Sign Your PGP Public Key
http://pgp.tnjc.edu.tw/pgp/FAQ.html	International PGP FAQ
http://www.cypherspace.org/~adam/hashcash/	Hash cash: digital cash
http://www.atthe.net/crypto.html	cryptography links
http://theory.lcs.mit.edu/~jacm/Authors/rivestronaldl.html	Ronald L. Rivest publications
http://www.cs.dartmouth.edu/FPCC/papers/	FPCC : Conference Proceedings
http://www.cert.org/	CERT Coordination Center
http://web.mit.edu/network/pgp.html	MIT distribution site for PGP
http://www.mathpuzzle.com/encrypt.htm	links to encryption information
http://www.rbjones.com/rbjpub/cs/sec001.htm	Security Net-Links
http://www.heureka.clara.net/sunrise/pgpwhat.htm	What is Pretty Good Privacy?
http://www.heureka.clara.net/sunrise/pgpwhy.htm	Why Use Pretty Good Privacy?
http://www.cl.cam.ac.uk/~fapp2/software/java-ssh/	Cédric Gourio's Java-SSH
http://www.ijs.si/ssh/	ssh: Secure Shell
http://www.stack.nl/~galactus/remailers/index-pgp.html	Security: Encryption: PGP
http://theory.lcs.mit.edu/~cis	Cryptography and Information Security Group (CIS Group) Home Page
http://www.scientific-computing.com/kenward/kwdoct98.html	Scientific Computing World

Table 4: Members of the "Ronald Rivest" community with the highest scores, sorted in descending order.

We believe that there are many applications for the methods that we have produced. Three of the more interesting applications are focused crawlers, automatic population of portal categories, and improved filtering.

Because the coverage of search engines is so poor, specialized search engines could use a focused crawler to exhaustively track a web community, thus increasing coverage of the community as well as increasing precision of search results.

For the second application, one could use existing portal categories as seeds to re-populate categories with newer and more relevant sites, thus addressing the lack of recall that portals are known to have and easing the burden on humans that manually construct such portals [24].

In terms of filtering, controversial web sites such as pornography and hate sites could also be identified; since pornography accounts for approximately 2% of the web [2], this is not out of the question. Moreover, link based community identification would not be fooled by language issues or keyword spamming.

Future work will generalize our notion of a community by parameterizing it with a coupling factor; low parameter settings will yield large but weakly connected communities, while high parameter values will find small, tightly connected communities.

We also believe that our maximum flow methods can be combined with other text-based methods in the form of co-learning and co-boosting. Text-based classifiers can be used to supply seeds and validate community membership; our flow method can generate new labels for training data; and the combined procedure can iterate, thus improving both methods.

6. ACKNOWLEDGMENTS

We would like to thank Kurt Bollacker, Frans Coetzee, Eric Glover, Ronitt Rubinfeld, Olga Veksler, and Leonid Zosin for many helpful discussions.

APPENDIX

A. ISA ALGORITHM

Most modern implementations of maximum flow algorithms rely on having access to the entire graph under consideration in order to make the flow analysis efficient. For example, the pre-flow push algorithm [25] (considered the fastest in practice) often uses a topological sort of all edges in order to improve efficiency. Clearly, global access to the web graph is not practical if one wishes to calculate an exact community.

On the other extreme, the shortest augmentation algorithm works by simply finding shortest paths between the source and the sink, augmenting flow along the discovered path, and repeats until the source and sink are disconnected. The problem with this approach is that it requires a breadth-first search (BFS) to be restarted from scratch after every augmentation.

Over the course of this research, we discovered a modification to the shortest augmentation algorithm that enables

```
procedure ISA(graph: G = (V, E); vertex: s, t ∈ V)
  Let Q be a BFS search queue
  while Q is not empty do
     if BFS search of G returns a shortest path then
        while label of t is valid do
           Augment flow along discovered path.
           Identify all vertices in BFS tree
             with invalid distance labels.
           Save invalid vertices in list, l.
           for v in l do
              Find best edge from v to a valid vertex.
              Update distance label of v.
           end for
           Sort vertices in l with a bin sort
             according to distance.
           for i equals 1 to maximum bin do
              for each v in bin i do
                 for all (v, u) ∈ E
                       with c(v, u) > 0 and u in l do
                    if dist. of u > dist. of v + 1 then
                       Relabel u with distance of v + 1.
                    end if
                 end for
              end for
           end for
           Reorder Q to reflect corrected distance labels.
        end while
     end if
end procedure
```

Table 5: The Incremental Shortest Augmentation (ISA) algorithm for solving the maximum flow problem.

the algorithm to retain as much as possible of the previous BFS search tree. The worst case runtime is identical to the standard shortest augmentation algorithm; however, the correction phase runs in time that is linear in the sum of the number of vertices with newly invalid distance labels and the outbound edges that those vertices have.

We refer to this flow algorithm as the *incremental shortest augmentation* (ISA) algorithm. ISA is described in pseudocode in Table 5. An intuitive description of the ISA follows.

Imagine that we build a physical model of a unit capacity undirected graph with strings and washers.[4] The vertices of the graph are represented by washers and the edges are represented by strings. Thus, for any two connected vertices, we tie the two corresponding washers together with a string. Since this is a unit capacity graph, all of the strings that connect washers will be the same distance, say six inches.

Place the "sink" washer on a table top and lock it in place with some tape. The ISA algorithm consists of alternating between the following two steps:

1. Grab the "source" washer and slowly raise it above the table until it can be raised no more. The taut string path from the source to the sink is a shortest path from the source to the sink.

2. Starting from the top and working down, snip each string (edge) that is part of the shortest path.

[4] ISA has been generalized to graphs with non-unit capacity directed edges; this is just the simplest example.

These two steps are repeated until the source washer is free of the table. When the source washer is completely free, the cut is complete. All washers that are still connected either directly or indirectly to the source washer are part of the community.

Since ISA "holds" the source washer at the level found in step 1 while the cuts are made in step 2, much of the distance information found at earlier steps is retained. While the cuts are being made, some washers may fall to the table, come free, stay in place, or fall only a short distance. When the final string is cut in step 2, the washers and strings are in a state identical to where they would have been had the path that was just removed never been in place. This result is identical to what is achieved by ISA's correction phase, which essentially imitates the effect of gravity on the washers. By way of comparison, other shortest augmentation path algorithms essentially restart step 1 with the source washer back on the table top, that is, the breadth first search restarts at the beginning.

ISA's correction phase operates in time $O(n' + m')$ where n' is the number of invalid vertices identified in lines 7 and 8 of Table 5, and m' is the number of edges connected to the n' invalid vertices.

B. REFERENCES

[1] Inktomi Corporation. Inktomi webmap press release. `http://www.inktomi.com/webmap/`, January 2000.

[2] Steve Lawrence and C. Lee Giles. Accessibility of information on the web. *Nature*, 400(6740):107–109, 1999.

[3] John Scott. *Social network analysis: a handbook.* SAGE Publications, 1991.

[4] E. Garfield. *Citation Indexing: Its Theory and Application in Science.* Wiley, New York, 1979.

[5] H.D. White and K.W. McCain. Bibliometrics. In *Ann. Rev. Info. Sci. and Technology*, pages 119–186. Elsevier, 1989.

[6] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Am. Soc. for Inf. Sci.*, 24(4):265–269, 1973.

[7] M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14:10–25, 1963.

[8] R. Larson. Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Ann. Meeting of the American Soc. Info. Sci.* 1996.

[9] S. Selim and M. Ismail. K-Means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):81–87, 1984.

[10] A. Popescul, G. W. Flake, S. Lawrence, L. Ungar, and C. L. Giles. Clustering and identifying temporal trends in document databases. In *Proc. IEEE Advances in Digital Libraries 2000*, 2000. To appear.

[11] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[12] L. Page. PageRank: bringing order to the web. Stanford Digital Libraries working paper 1997-0072., 1997.

[13] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proc. 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[14] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* W. H. Freeman, New York, 1979.

[15] C. Chekuri, A. Goldberg, D. Karger, M. Levine, and C. Stein. Experimental study of minimum cut algorithms. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 324–333, New Orleans, 1997.

[16] L. R. Ford Jr. and D. R. Fulkerson. Maximal flow through a network. *Canadian J. Math.*, 8:399–404, 1956.

[17] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms.* MIT Press and McGraw-Hill Book Company, 6th edition, 1992.

[18] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows : Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[19] J. Edmonds and R. M. Karp. Theoretical improvements in the algorithmic efficiency for network flow problems. *JACM*, 19:248–264, 1972.

[20] R. Albert, H. Jeong, and A.-L. Barabasi. Diameter of the world wide web. *Nature*, 401, 1999.

[21] S. Chakrabarti, M. van der Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Proceedings of 8th International World Wide Web Conference (WWW8)*, 1999.

[22] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *PAMI*, 15(11):1101–1113, November 1993.

[23] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:185–197, 1977.

[24] S. Macskassy, A. Banerjee, B. Davison, and H. Hirsh. Human performance on clustering web pages: A preliminary study. In *Proceedings of The Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, 1998.

[25] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum flow problem. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 136–146, Berkeley, California, 28–30 May 1986.