Web Structure Mining
Exploiting the Graph Structure of the World-Wide Web

Johannes Fürnkranz
Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Wien, Austria
E-mail: juffi@oefai.at

Abstract

The World-Wide Web provides every internet citizen with access to an abundance of information, but it becomes increasingly difficult to identify the relevant pieces of information. Web mining is a new research area that tries to address this problem by applying techniques from data mining and machine learning to Web data and documents. In this paper, we will give a brief overview of Web mining, with a special focus on techniques that aim at exploiting the graph structure of the Web for improved retrieval performance and classification accuracy.

1 Web Mining

The advent of the World-Wide Web (WWW) has overwhelmed the typical home computer user with an enormous flood of information. To almost any topic one can think of, one can find pieces of information that are made available by other internet citizens, ranging from individual users that post an inventory of their record collection, to major companies that do business over the Web.

To be able to cope with the abundance of available information, users of the WWW need to rely on intelligent tools that assist them in finding, sorting, and filtering the available information. Just as data mining aims at discovering valuable information that is hidden in conventional databases, the emerging field of *Web mining* aims at finding and extracting relevant information that is hidden in Web-related data, in particular in text documents that are published on the Web. Like data mining, Web mining is a multi-disciplinary effort that draws techniques from fields like information retrieval, statistics, machine learning, natural language processing, and others.

Depending on the nature of the data, one can distinguish three main areas of research within the Web mining community:

Web Content Mining: application of data mining techniques to unstructured or semi-structured data, usually HTML-documents

Web Structure Mining: use of the hyperlink structure of the Web as an (additional) information source

Web Usage Mining: analysis of user interations with a Web server (e.g., click-stream analysis)

For an excellent survey of the field, cf. (Chakrabarti, 2000), a slim textbook appeared as (Chang et al., 2001). For a survey of content mining, we refer to (Sebastiani, 2002), while a survey of usage mining can be found in (Srivastava et al., 2000). We are not aware of a previous survey on structure mining.

2 Motivation

2.1 The Web is a Graph

While conventional information retrieval focuses primarily on information that is provided by the text of Web documents, the Web provides additional information through the way in which different documents are connected to each other via *hyperlinks*. The Web may be viewed as a (directed) graph whose nodes are the documents and the edges are the hyperlinks between them.

Several authors have tried to analyze the properties of this graph. The most comprehensive study is due to Broder et al. (2000). They used data from an Altavista crawl (May 1999) with 203 million URLs and 1466 million links, and stored the underlying graph structure in a connectivity server (Bharat et al., 1998), which implements an efficient document indexing technique that allows fast access to both outgoing and incoming hyperlinks of a page. The entire graph fitted in 9.5 GB of storage, and a breadth-first search that reached 100M nodes took only about 4 minutes.

Their main result is an analysis of the structure of the web graph, which, according to them, looks like giant bow tie, with a strongly connected core component (SCC) of 56 million pages in the middle, and two components with 44 million pages each on the sides, one containing pages from which the SCC can be reached (the IN set), and the other containing pages that can be reached from the SCC (the OUT set). In addition, there are "tubes" that allow to reach the OUT set from the IN set without passing through the SCC, and many "tendrils", that lead out of the IN set or into the OUT set with connecting to other components. Finally, there are also several smaller components that cannot be reached from any point in this structure. (Broder et al., 2000) also sketch a diagram of this structure, which is somewhat deceptive because the prominent role of the IN, OUT, and SCC sets is based on size only, and there are other structures with a similar shape, but of somewhat smaller size (e.g., the tubes may contain other strongly connected components that differ from the SCC only in size). The main result is that there are several disjoint components. In fact, the probability that a path between two randomly selected pages exists is only about 0.24.

Based on the analysis of this structure, Broder et al. (2000) estimated that the diameter (i.e., the maximum of the lengths of the shortest paths between two nodes) of the SCC is larger than 27, that the diameter of the entire graph is larger than 500, and that the average length of such a path is about 16. This is, of course only for cases where a path between two pages exists. These results correct earlier estimates obtained by Albert et al. (1999) who estimated the average length at about 19. Their analyis was based on a probabilistic argument using estimates for the in-degrees and out-degrees, thereby ignoring the possibility of disjoint components.

Albert et al. (1999) base their analysis on the observation that the in-degrees

(number of incoming links) and out-degrees (number of outgoing links) follow a power law distribution $P(d) \approx d^{-\gamma}$. They estimated values of $\gamma_{in} = 2.45$ and $\gamma_{out} = 2.1$ for the in-degrees and out-degrees respectively. They also note that these power law distributions imply a much higher probability of encountering documents with large in- or out-degrees than would be the case for random networks or random graphs. The power-law results have been confirmed by Broder et al. (2000) who also observed a power law for the sizes of strongly connected components in the Web graph. Faloutsos et al. (1999) observed a Zipf distribution $P(d) \approx r(d)^{-\gamma}$ of the outdegree of nodes, where $r(d)$ is the rank of the degree in a sorted list of out-degree values. Similarly, Levene et al. (2001) observed a Zipf distribution in a model of the behavior of Web surfers.

Finally, another interesting property is the size of the Web. Lawrence and Giles (1998) suggest to estimate the size of the Web from the overlap that different search engines return for identical queries. Their method is based on the assumption that the probability that a page is indexed by search engine A is independent of the probability that this page is indexed by search engine B. In this case, the percentage of pages in the result set of a query for search engine B that are also indexed by search engine A could be used as an estimate for the over-all percentage of pages indexed by A. Obviously, the independence assumption on which this argument is based does not hold in practice, so that the estimated percentage is larger than the real percentage (and the obtained estimates of the Web size are more like lower bounds). Lawrence and Giles (1998) used the results of several queries to estimate that the largest search engine indexes only about one third of the indexable Web (the portion of the Web that is accessible to crawlers, i.e., not hidden behind query interfaces). Similar arguments were used by Bharat and Broder (1998) to estimate the relative size of search engines.

2.2 The Importance of Predecessor Pages

In the following, we try to motivate that the information on *predecessor pages*—pages that have a hyperlink pointing to the *target page*—may contain particularly useful information for classifying a page:

redundancy: Quite often there is more than one page pointing to a single page on the Web. The ability to combine multiple, independent sources of information can improve classification accuracy as is witnessed by the success of ensemble techniques in other areas of machine learning (Dietterich, 2000).

independent encoding: As the set of predecessor pages typically originates from several different authors, the provided information is less sensitive to the vocabulary used by one particular author.

sparse or non-existing text: Many Web pages only contain a limited amount of text. In fact, many pages only contain images and no machine-readable text at all. Looking at predecessor pages increases the chances of encountering informative text about the page to classify.

irrelevant or misleading text: Often pages contain irrelevant text. In other cases, pages are designed to deliberately contain misleading information. For example, many pages try to include a lot of keywords into comments

or invisible parts of the text in order to increase the breadth of their indexing for word-based search engines.[1] Again, the fact that predecessor pages are typically authored by different and independent sources may provide relevant information or improve focus.

foreign language text: English is (still) the predominant language on the Web. Nevertheless, documents in other languages occur in non-negligible numbers. Even if the text on the page is written in a foreign language, there may be incoming links from pages that are written in English. In many cases, this allows an English speaker (or a text classifier based on English vocabulary) to infer something about the contents of the page.

In summary, the use of text on predecessor pages may provide a richer vocabulary, independent assessment of the contents of a page through multiple authors, redundancy in classification, focus on important aspects, and a simple mechanism for dealing with pages that have sparse text, no text, or text in different languages.

3 Making Use of the Graph Structure

3.1 Retrieval

The importance of information contained in the hyperlinks pointing to a page has been recognized early on. Anchor texts (texts on hyperlinks in an HTML document) of predecessor pages were already indexed by the World-Wide Web Worm, one of the first search engines and Web crawlers (McBryan, 1994). Spertus (1997) suggested a taxonomy of different types of (hyper-)links that can be found on the Web and discusses how the links can be exploited for various information retrieval tasks on the Web.

However, the main break-through was the realization that the popularity and hence the importance of a page is—to some extent—correlated to the number of incoming links, and that this information can be adventageously used for sorting the query results of a search engine. The in-degree alone, however, is a poor measure of importance because many pages are frequently pointed to without being connected to the contents of the referring page (think, e.g., the numerous "best viewed with..." hyperlinks that point to browser home-pages). More sophisticated measures are needed.

Kleinberg (1999) realized that there are two types of pages that could be relevant for a query: *authorities* are pages that contain useful information about the query topic, while *hubs* contain pointers to good information sources. Obviously, both types of pages are typically connected: good hubs contain pointers to many good authorities, and good authorities are pointed to by many good hubs. Kleinberg (1999) suggested to make practical use of this relationship by associating each page x with a hub score $H(x)$ and an authority score $A(x)$, which are computed iteratively:

[1] For example, the author once encountered a page of an Austrian ski ressort that contained a complete list of Austrian ski ressorts in comments, apparently with the goal of being retrieved if somebody looks for information of one of its competitors (in fact this was how the page was found by the author). For recognizing this page as a ski ressort, this information may be helpful. On the other hand, the author also came across a page that included an entire dictionary of the most common words of the English language as invisible text...

$$H_{i+1}(x) = \sum_{(x,s)} A_i(s)$$
$$A_{i+1}(x) = \sum_{(p,x)} H_i(p)$$

where (x, y) denotes that there is a hyperlink from page x to page y. This computation is conducted on a so-called *focussed subgraph* of the Web, which is obtained by enhancing the search result of a conventional query (or a bounded subset of the result) with all predecessor and successor pages (or, again, a bounded subset of them). The hub and authority scores are initilialized uniformly with $A_0(x) = H_0(x) = 1.0$ and normalized so that they sum up to one before each iteration. Kleinberg (1999) was able to prove that this algorithm (called HITS) will always converge, and practical experience shows that it will typically do so within a few iterations (about 5; Chakrabarti et al., 1998b). HITS has been used for identifying relevant documents for topics in web catalogues (Chakrabarti et al., 1998b; Bharat and Henzinger, 1998) and for implementing a "Related Pages" functionality (Dean and Henzinger, 1999).

The main drawback of the HITS algorithm is that the hubs and authority score must be computed iteratively from the query result, which does not meet the real-time constraints of an on-line search engine. However, the implementation of a similar idea in the Google search engine resulted in a major break-through in search engine technology. Brin and Page (1998) suggested the use of the probability that a page is visited by a random surfer on the Web as a key factor for ranking search results. They approximated this probability with the so-called *PageRank*, which is again computed iteratively:

$$PR_{i+1}(x) = (1 - l)\frac{1}{N} + l \sum_{(p,x)} \frac{PR_i(p)}{|(p, y)|} \tag{1}$$

The first term of this sum models the behavior that a surfer gets bored (with probability $(1 - l)$, where l is typically set to 0.85) and jumps to a randomly selected page of the entire set of N pages. The second term uniformly distributes the current page rank of a page to all its successor pages. Thus, a page receives a high page rank if it is linked by many pages, which in turn have a high page rank and/or only few successor pages. The main advantage of the page rank over the hubs and authority scores is that it can be computed off-line, i.e., it can be precomputed for all pages in the index of a search engine. Its clever (but secret) integration with other information that is typically used by search engines (number of matching query terms, location of matches, proximity of matches, etc.) promoted Google from a student project to the main player in search engine technology.

3.2 Classification

Not surprisingly, recent research has also looked at the potential of hyperlinks as additional information source for hypertext categorization tasks. Many authors addressed this problem in one way or another by merging (parts of) the text

of the predecessor pages with the text of the page to classify, or by keeping a separate feature set for the predecessor pages. For example, Chakrabarti et al. (1998a) evaluated two variants, one that simply appends the text of the neighboring (predecessor and successor) pages to the text of the target page, and one that uses two different sets of features, one for the target page and one for a concatenation of the neighboring pages. The results were negative: in two domains both approaches performed worse than the conventional technique that uses only features of the target document. From these results, Chakrabarti et al. (1998a) concluded that the text from the neighbors is too noisy to help classification and proposed a different technique that included predictions for the class labels of the neighboring pages into the model. Unless the labels for the neighbors are known a priori, the implementation of this approach requires an iterative technique for assigning the labels, because changing the class of a page may potentially change the class assignments for all neighboring pages as well. The authors implemented a relaxation labeling technique, and showed that it improves performance over the standard text-based approach that ignores the hyperlink structure. The utility of classes predictions for neighboring pages was confirmed by the results of Oh et al. (2000) and Yang et al. (2002).

A different line of research concentrates on explicitly encoding the relational structure of the Web in first-order logic. For example, a binary predicate `link_to(page1,page2)` can be used to represent the fact that there is a hyperlink on `page1` that points to `page2`. In order to be able to deal with such a representation, one has to go beyond traditional attribute-value learning algorithms and resort to inductive logic programming, aka relational data mining (Džeroski and Lavrač, 2001). Craven et al. (1998) use a variant of Foil (Quinlan, 1990) to learn classification rules that can incorporate features from neighboring pages. The algorithm uses a deterministic version of relational path-finding (Richards and Mooney, 1992), which overcomes Foil's restriction to determinate literals (Quinlan, 1991), to construct chains of `link_to/2` predicates that allow the learner to access the words on a page via a predicate of the type `has_word(page,word)`. For example, the conjunction `link_to(P1,P), has_word(P1,word)` means "there exists a predecessor page `P1` that contains the word `word`. Slattery and Mitchell (2000) improve the basic Foil-like learning algorithm by integrating it with ideas originating from the HITS algorithm for computing hub and authority scores of pages, while Craven and Slattery (2001) combine it favorably with a Naive Bayes classifier.

At its core, using features of pages that are linked via a `link_to/2` predicate is quite similar to the approach evaluated by Chakrabarti et al. (1998a) where words of neighboring documents are added as a separate feature set: in both cases, the learner has access to all the features in the neighboring documents. The main difference lies in the fact that in the relational representation, the learner may control the depth of the chains of `link_to/2` predicates, i.e., it may incorporate features from pages that are several clicks apart. From a practical point of view, the main difference lies in the types of learning algorithms that are used with both approaches: while inductive logic programming typically relies on rule learning algorithms which classify pages with "hard" classification rules that predict a class by looking only at a few selected features, Chakrabarti et al. (1998a) used learning algorithms that always take all available features into account (such as a Naive Bayes classifier). Yang et al. (2002) discuss both approaches, relate them to a taxonomy of five possible regularities that may be

present in the neighborhood of a target page, and empirically compare them under different conditions.

3.3 Classification with Hyperlink Ensembles

However, the above-mentioned approaches still suffer from several short-comings:

- *Features of predecessor pages should be kept separately:* The failed attempt by Chakrabarti et al. (1998a) merges the entire text from all predecessor pages into a single pot.
- *Redundancy provided by multiple predecessors should be exploited:* ILP approaches can (in principle) keep features separately, but are restricted to the use of existential variables in the rules (i.e., they can only formulate rules of the type "there exists a predecessor page with such and such properties".
- *Not the entire text of a predecessor page is relevant:* Each page is predecessor of several pages, each of which may belong to a different class. Thus, if the entire text is used, each outgoing hyperlink is represented in the same way. On the other hand, simply focussing on the anchor text, is too narrow.
- *Not all pages have relevant meta-information:* The solution proposed by Chakrabarti et al. (1998a) assumes that each predecessor page of a page is part of the classification problem, i.e., it can be assigned to a meaningful category. This need not be the case.

In (Fürnkranz, 1999; 2001), we introduced the use of *hyperlink ensembles* for classification of hypertext pages, which address the above-mentioned problems. The idea is quite simple: instead of training a classifier that classifies *pages* based on the words that appear in their text, we propose to train a classifier that classifies *hyperlinks* according to the class of the pages they point to, based on the words that occur in their neighborhood of the link (in the simplest case the anchor text of the link). Consequently, each page will be assigned multiple predictions for its class membership, one for each incoming hyperlink. These individual predictions are then combined to a final prediction by some voting procedure. Thus, the technique is a member of the family of ensemble learning methods (Dietterich, 2000).

In a preliminary empirical evaluation in the WebKB domain (where the task is to recognize typical entities in Computer Science departments, such as faculty, student, course, and project pages; Craven et al. 2000), hyperlink ensembles outperformed a conventional full-text classifier. We also studied the effect of different voting schemes and, more importantly, different feature representations. In addition to the use of anchor text, we also investigated the use of headings preceding the current paragraph, the use of the text in the current paragraph, and the use of linguistic phrases that occur in the current paragraph. The combination of anchor text and heading features turned out to be most important, but the addition of paragraph text and phrasal features still lead to minor improvements. The main problem with phrasal features with the phrasal features is that they are too specific to be of general use (low recall; cf. also

Fürnkranz et al., 1998). The overall classifier improved the full-text classifier from about 70% accuracy to about 85% accuracy in this domain. It is still open to see whether this generalizes to other domains.

4 Conclusion

The main purpose of this paper was to motivate that the graph structure of the Web may provide a valuable source of information for various Web mining tasks, as witnessed by the success of search engines that try to incorporate graph properties into the ranking of their query results. In particular, we wanted to show that the information of predecessor pages can be successively used for improving the performance of text classification tasks. We reviewed various previous approaches, and briefly discussed our own solution. However, research in this area has just begun...

Acknowledgments

The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture.

References

R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the world-wide web. *Nature*, 401:130–131, September 1999.

K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. *Computer Networks*, 30(1–7):107–117, 1998. Proceedings of the 7th International World Wide Web Conference (WWW-7), Brisbane, Australia.

K. Bharat, A. Broder, M. R. Henzinger, P. Kumar, and S. Venkatasubramanian. The connectivity server: Fast access to linkage information on the Web. *Computer Networks*, 30(1–7):469–477, 1998. Proceedings of the 7th International World Wide Web Conference (WWW-7), Brisbane, Australia.

K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98)*, pages 104–111, 1998.

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30(1–7):107–117, 1998. Proceedings of the 7th International World Wide Web Conference (WWW-7), Brisbane, Australia.

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. *Computer Networks*, 33(1–6):309–320, 2000. Proceedings of the 9th International World Wide Web Conference (WWW-9).

S. Chakrabarti. Data mining for hypertext: A tutorial survey. *SIGKDD explorations*, 1(2):1–11, January 2000.

S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD International Conference on Management on Data*, pages 307–318, Seattle, WA, 1998a. ACM Press.

S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks*, 30(1–7):65–74, 1998b. Proceedings of the 7th International World Wide Web Conference (WWW-7), Brisbane, Australia.

G. Chang, M. J. Healy, J. A. M. McHugh, and J. T. L. Wang. *Mining the World Wide Web: An Information Search Approach*. Kluwer Academic Publishers, 2001.

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1-2):69–114, 2000.

M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1-2):97–119, 2001.

M. Craven, S. Slattery, and K. Nigam. First-order learning for Web mining. In C. Nédellec and C. Rouveirol, editors, *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, pages 250–255, Chemnitz, Germany, 1998. Springer-Verlag.

J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In A. Mendelzon, editor, *Proceedings of the 8th International World Wide Web Conference (WWW-8)*, pages 389–401, Toronto, Canada, 1999.

T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000.

S. Džeroski and N. Lavrač, editors. *Relational Data Mining: Inductive Logic Programming for Knowledge Discovery in Databases*. Springer-Verlag, 2001.

M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM-99)*, pages 251–262, Cambridge, MA, 1999. ACM Press.

J. Fürnkranz. Exploiting structural information for text classification on the WWW. In D. Hand, J. N. Kok, and M. Berthold, editors, *Advances in Intelligent Data Analysis: Proceedings of the 3rd International Symposium (IDA-99)*, pages 487–497, Amsterdam, Netherlands, 1999. Springer-Verlag.

J. Fürnkranz. Hyperlink ensembles: A case study in hypertext classification. Technical Report OEFAI-TR-2001-30, Austrian Research Institute for Artificial Intelligence, Wien, Austria, 2001. Accepted pending revisions for *Information Fusion*, Special Issue on Fusion of Multiple Classifiers.

J. Fürnkranz, T. Mitchell, and E. Riloff. A case study in using linguistic phrases for text categorization on the WWW. In M. Sahami, editor, *Learning for Text Categorization: Proceedings of the 1998 AAAI/ICML Workshop*, pages 5–12, Madison, WI, 1998. AAAI Press. Technical Report WS-98-05.

J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999.

S. Lawrence and C. L. Giles. Searching the world wide web. *Science*, 280: 98–100, 1998.

M. Levene, J. Borges, and G. Louizou. Zipf's law for Web surfers. *Knowledge and Information Systems*, 3(1):120–129, 2001.

O. A. McBryan. GENVL and WWWW: Tools for taming the Web. In *Proceedings of the 1st World-Wide Web Conference (WWW-1)*, pages 58–67, Geneva, Switzerland, 1994. Elsevier.

H.-J. Oh, S. H. Myaeng, and M.-H. Lee. A practical hypertext categorization method using links and incrementally available class information. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR-00)*, pages 264–271, Athens, Greece, 2000.

J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

J. R. Quinlan. Determinate literals in inductive logic programming. In *Proceedings of the 8th International Workshop on Machine Learning (ML-91)*, pages 442–446, 1991.

B. L. Richards and R. J. Mooney. Learning relations by pathfinding. In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, pages 50–55, San Jose, CA, 1992. AAAI Press.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

S. Slattery and T. Mitchell. Discovering test set regularities in relational domains. In P. Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML-00)*, pages 895–902, Stanford, CA, 2000. Morgan Kaufmann.

E. Spertus. ParaSite: Mining structural information on the Web. *Computer Networks and ISDN Systems*, 29(8-13):1205–1215, September 1997. Proceedings of the 6th International World Wide Web Conference (WWW-6).

J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD explorations*, 1(2):12–23, 2000.

Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2–3):219–241, March 2002. Special Issue on Automatic Text Categorization.