Chapter #

WEB MINING

Johannes Fürnkranz
TU Darmstadt, Knowledge Engineering Group

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. Research in web mining tries to address this problem by applying techniques from data mining and machine learning to Web data and documents. This chapter provides a brief overview of web mining techniques and research areas, most notably hypertext classification, wrapper induction, recommender systems and web usage mining.

Key words: web mining, content mining, structure mining, usage mining, text classification, hypertext classification, information extraction, wrapper induction, collaborative filtering, recommender systems, Semantic Web

1. INTRODUCTION

The advent of the World-Wide Web (WWW) (Berners-Lee, Cailliau, Loutonen, Nielsen & Secret, 1994) has overwhelmed home computer users 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 Web need assistance of intelligent software agents (often called *softbots*) for finding, sorting, and filtering the available information (Etzioni, 1996; Kozierok & Maes, 1993). Beyond search engines, which are already commonly used, research concentrates on the development of agents that are general, high-level interfaces to the Web (Etzioni & Weld, 1994; Fürnkranz, Holzbaur & Temel, 2002), programs for filtering and sorting e-mail messages (Maes, 1994; Payne & Edwards, 1997) or Usenet netnews articles (Lashkari, Metral & Maes, 1994; Sheth & Maes, 1993; Lang, 1995; Mock, 1996), recommender systems for suggesting Web sites (Armstrong, Freitag, Joachims & Mitchell, 1995; Pazzani, Muramatsu & Billsus, 1996; Balabanovic,´ & Shoham, 1995) or products (Doorenbos, Etzioni & Weld, 1997; Burke, Hammond & Young, 1996), automated answering systems (Burke, Hammond, Kulyukin, Lytinen, Tomuro & Schoenberg, 1997; Scheffer, 2004) and many more.

Many of these systems are based on machine learning and data mining techniques. 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 (hyper-)text documents 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.

Web mining is commonly divided into the following three sub-areas:

- **Web Content Mining:** application of data mining techniques to unstructured or semi-structured text, typically HTML-documents
- **Web Structure Mining:** use of the hyperlink structure of the Web as an (additional) information source
- **Web Usage Mining:** analysis of user interactions with a Web server

An excellent textbook for the field is (Chakrabarti, 2002), an earlier effort (Chang, Healy, McHugh & Wang, 2001). Brief surveys can be found in (Chakrabarti, 2000; Kosala & Blockeel, 2000). For surveys of content mining, we refer to (Sebastiani, 2002), while a survey of usage mining can be found in (Srivastava, Cooley, Deshpande & Tan, 2000). We are not aware of a previous survey on structure mining.

In this chapter, we will organize the material somewhat differently. We start with a brief introduction on the Web, in particular on its unique properties as a graph (Section 2), and subsequently discuss how these properties are exploited for improved retrieval performance in search engines (Section 3). After a brief recapitulation of text classification (Section 4), we discuss approaches that attempt to use the link structure of the Web for improving hypertext classification (Section 5). Subsequently, we summarize important research in the areas information extraction and wrapper induction (Section 6), and briefly discuss the web mining opportunities of the Semantic Web (Section 7). Finally, we present research in web usage mining (Section 8) and recommender systems (Section 9).

2. GRAPH PROPERTIES OF THE WEB

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 with documents as nodes and hyperlinks as edges.

Several authors have tried to analyze the properties of this graph. The most comprehensive study is due to (Broder, Kumar, Maghoul, Raghavan, Rajagopalan, Stata 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, Broder, Henzinger, Kumar & Venkatasubramanian, 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 without 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, Jeong, and Barabási (1999) who estimated the average length at about 19. Their analysis 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 γ_{in}=2.45 and γ_{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 distribution for the sizes of strongly connected components in the web graph. Faloutsos, Faloutsos & Faloutsos (1999) observed a Zipf distribution $P(d) \approx r(d)^{-\gamma}$ for the out-degree of nodes ($r(d)$ is the rank of the degree in a sorted list of out-degree values). Similarly, a model of the behavior of web surfers was shown to follow a Zipf distribution (Levene, Borges & Louizou, 2001).

Finally, another interesting property is the size of the Web. Lawrence and Giles (1998) propose 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.

3. WEB SEARCH

Whereas conventional query interfaces concentrate on indexing documents by the words that appear in them (Salton, Wong & Yang, 1975; Salton, 1989), the potential of utilizing the 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) introduced a taxonomy of different types of (hyper-)links that can be found on the Web, and discussed 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 advantageously 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) suggest that 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) suggest 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:

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

where (x,y) denotes that there is a hyperlink from page x to page y. This computation is conducted on a so-called *focused 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 initialized uniformly with $A_0(x)=H_0(x)=1.0$ and normalized so that they sum up to one before each iteration. It can be proven that this algorithm (called HITS) will always converge (Kleinberg, 1999), and practical experience shows that it will typically do so within a few iterations (about 5; Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson & Kleinberg, 1998b). Variants of the HITS algorithm have been used for identifying relevant documents for topics in web catalogues (Chakrabarti et al., 1998b; Bharat & Henzinger, 1998) and for implementing a “Related Pages” functionality (Dean & Henzinger, 1999).

The main drawback of this 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 & Page, 1998). The key idea is to use the probability that a page is visited by a random surfer on the Web as an important factor for ranking search results. This probability is approximated by the so-called *page rank*, which is again computed iteratively:

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

The first term of this sum models the behavior that a surfer gets bored and jumps to a randomly selected page of the entire set of *N* pages (with probability (1-*l*), where *l* is typically set to 0.85). 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.

4. TEXT CLASSIFICATION

Text classification is the task of sorting documents into a given set of categories. One of the most common web mining tasks is the automated induction of such text classifiers from a set of training documents for which the category is known. A detailed overview of this field can be found in (Sebastiani, 2002), as well as in the corresponding Chapter of this book. The main problem, in comparison to conventional classification tasks, is the additional degree of freedom that results from the need to extract a suitable feature set for the classification task. Typically, each word is considered as a separate feature with either a Boolean value indicating whether the word occurs or does not occur in the document (*set-of-words* representation) or a numeric value that indicates the frequency (*bag-of-words* representation). A comparison of these two basic models can be found in (McCallum & Nigam, 1998). Advanced approaches use different weights for terms (Salton & Buckley, 1988), more elaborate feature sets like *n*-grams (Mladenić & Grobelnik, 1998; Fürnkranz, 1998) or linguistic features (Lewis, 1992; Fürnkranz, Mitchell & Riloff, 1998; Scott & Matwin, 1999), linear combinations of features (Deerwester, Dumais, Landauer, Furnas & Harshman, 1990) or rely on automated feature selection techniques (Yang & Pedersen, 1997; Mladenić, 1998a).

There are numerous application areas for this type of learning task (Mladenić, 1999). For example, the generation of web catalogues such as `http://www.dmoz.org/` is basically a classification task that assigns documents to labels in a structured hierarchy of classes. Typically, this task is performed manually by a large user community or employees of companies that specialize in such efforts, like Yahoo!. Automating this assignment is a rewarding task for text categorization and text classification (Mladenić, 1998b).

Similarly, the sorting of one's personal E-mail messages into a flat or structured hierarchy of mail folders is a text categorization task that is mostly performed manually, sometimes supported with manually defined classification rules. Again, there have been numerous attempts in augmenting this procedure with automatically induced content-based classification rules (Cohen, 1996; Payne & Edwards, 1997; Crawford, Kay & McCreath, 2002). Recently, a related task has received increased attention, namely automated filtering of spam mail. Training classifiers for recognizing spam mail is a particularly challenging problem for machine learning, involving skewed example distributions, misclassification costs, concept drift, undefined feature sets, and more (Fawcett, 2003). Most algorithms, such as the built-in spam filter of the Mozilla open source browser (Graham, 2003), rely on Bayesian learning for tackling this problem. A comparison of different learning algorithms for this problem can be found in (Androutsopoulos, Paliouras & Michelakis, 2004).

5. HYPERTEXT 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, Dom, and Indyk (1998a) evaluate two variants: (1) appending the text of the neighboring (predecessor and successor) pages to the text of the target page, and (2) using 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. Chakrabarti et al. (1998a) concluded that the text from the neighbors is too unreliable to help classification. Consequently, a different technique was proposed 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 class predictions for neighboring pages was confirmed by the results of Oh, Myaeng, and Lee (2000) and Yang, Slattery, and Ghani (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 & Lavrač, 2001). Craven, Slattery & Nigam (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 & 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 in (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 characteristics of the used learning algorithms: 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 and relate them to a taxonomy of five possible regularities that may be present in the neighborhood of a target page. They also experimentally compare these approaches under different conditions.

However, the above-mentioned approaches still suffer from several shortcomings, most notably that only portions of the predecessor pages are relevant, and that not all predecessor pages are equally relevant. A solution attempt is provided by the use of *hyperlink ensembles* for classification of hypertext pages (Fürnkranz, 2002). The idea is quite simple: instead of training a classifier that classifies *pages* based on the words that appear in their text, a classifier is trained 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 Web→KB domain (where the task is to recognize typical entities in Computer Science departments, such as faculty, student, course, and project pages, cf. Section 7), hyperlink ensembles outperformed a conventional full-text classifier in a study that employed a variety of voting schemes for combining the individual classifiers and a variety of feature extraction techniques for representing the information around an incoming hyperlink (e.g., the anchor text on a hyperlink, the text in the sentence that contains the hyperlink, or the text of an entire paragraph). The overall classifier improved the full-text classifier from about 70% accuracy to about 85% accuracy in this domain. It remains to be seen whether this generalizes to other domains.

6. INFORMATION EXTRACTION AND WRAPPER INDUCTION

Information extraction is concerned with the extraction of certain information items from unstructured text. For example, you might want to extract the title, show times, and prices from web pages of movie theaters near you. While web search can be used to find the relevant pages, information extraction is needed to identify these particular items on each page. An excellent survey of the field can be found in (Eikvil, 1999). Premier events in this field include the *Message Understanding Conferences (MUC)*, and numerous workshops devoted to special aspects of this topic (Califf, 1999; Pazienza, 2003).

Information extraction has a long history. There are numerous algorithms that work with unstructured textual documents, mostly employing natural language processing. A typical system is AutoSlog (Riloff, 1996b), which was developed as a method for automatically constructing domain-specific extraction patterns from an annotated training corpus. As input, AutoSlog requires a set of noun phrases that constitute the information that should be extracted from the training documents. AutoSlog then uses syntactic heuristics to create linguistic patterns that can extract the desired information from the training documents (and from unseen documents). The extracted patterns typically represent subject–verb or verb–direct-object relationships (e.g., <*subject*> *teaches* or *teaches* <*direct-object*>) as well as prepositional phrase attachments (e.g., *teaches at* <*noun-phrase*> or *teacher at* <*noun-phrase*>). An extension, AutoSlog-TS (Riloff, 1996a) removes the need for an annotated training corpus by generating extraction patterns for *all* noun phrases in the training corpus whose syntactic role matches one of the syntactic heuristics.

Other systems that work with unstructured text are based on inductive rule learning algorithms that can make use of a multitude of features, including linguistic tags, HTML tags, font size, etc., and learn a set of extraction rules that specify which combination of features indicates an appearance of the target information. WHISK (Soderland, 1999) and SRV (Freitag, 1998) employ a top-down, general-to-specific search for finding a rule that covers a subset of the target patterns, whereas RAPIER (Califf, 2003) employs a bottom-up search that successively generalizes a pair of target patterns.

While the above-mentioned systems typically work on unstructured or semi-structured text, a new direction focused on the extraction of items from structured HTML-pages. Such *wrappers* identify their content primarily via a sequence of HTML tags (or an XPath in a DOM-tree). Kushmerick (2000) first studied the problem of inducing such wrappers from a set of training examples where the information to extract is marked. He studies a variety of types of wrapper algorithms with different expressiveness. The simplest class, LR wrappers, assume a highly regular source page that allows to map its content into a database table by learning delimiters for each attribute. LR wrappers were able to wrap 53% of the pages in an experimental study, more expressive classes were able to wrap up to 70%. Moreover, it was shown that all studied wrapper classes are PAC-learnable. Grieser, Jantke, Lange & Thomas (2000) extend this work with a study of theoretical properties and learnability results for island wrappers, a generalization of the wrapper types studied by Kushmerick (2000). SoftMealy (Hsu & Dung, 1998) addresses several of the short-comings of the framework of Kushmerick (2000), most notably the restriction to single sequences of features, by learning a finite-state transducer that allows to encode all occurring sequences of features.

Lerman, Minton, and Knoblock (2003) discuss learning approaches for supporting the maintenance of existing wrappers.

The field has also seen numerous commercial efforts, such as the Lixto project (Gottlob, Koch, Baumgartner, Herzog & Flesca, 2004) or IBM's Andes project (Myllymaki, 2001). The most notable application of information extraction techniques are comparison shopping agents (Doorenbos et al., 1997).

7. THE SEMANTIC WEB

The Semantic Web is a term coined by Tim Berner-Lee for the vision of making the information on the Web machine-processable (Berners-Lee, Hendler & Lassila, 2001). The basic idea is to enrich web pages with machine-processable knowledge that is represented in the form of *ontologies* (Fensel, 2001). Ontologies define certain types of objects and the relations between them. As ontologies are readily accessible (like other web documents), a computer program can use them to draw inferences about the information provided on web pages.

One of the research challenges in that area is to annotate the information that is currently available on the Web with semantic tags. Typically, techniques from text classification, hyper-text classification and information extraction are used for that purpose. A landmark application in this area was the Web→KB project at Carnegie-Mellon University (Craven, DiPasquo, Freitag, McCallum, Mitchell, Nigam & Slattery, 2000). Its goal was to assign web pages or parts of web pages to entities in an ontology. A simple test ontology modeled knowledge about computer science departments: there are entities like students (graduate and undergraduate), faculty members (professors, researchers, lecturers, post-docs, ...), courses, projects, etc., and relations between these entities, such as "courses are taught by one lecturer and attended by several students" or "every graduate student is advised by a professor". Many applications could be imagined for such an ontology. For example, it could enhance the capabilities of search engines by enabling them to answer queries like "Who teaches course *X* at university *Y*? " or "How many students are in department *Z*? ", or serve as a backbone for web catalogues (Staab & Maedche, 2001). A description of the first prototype system can be found in (Craven et al., 2000).

Semantic Web Mining emerged as research field that focuses on the interactions of web mining and the Semantic Web (Berendt, Hotho & Stumme, 2002). On the one hand, web mining can support the learning of ontologies in various ways (Maedche & Staab, 2001; Maedche, Pekar & Staab, 2003; Doan, Madhavan, Dhamankar, Domingos & Halevy, 2003). On the other hand, background knowledge in the form of ontologies may be used for supporting web mining tasks. Several workshops have been devoted to these topics (Staab, Maedche, Nédellec & Wiemer-Hastings, 2000; Maedche, Nédellec, Staab & Hovy, 2001; Stumme, Hotho & Berendt, 2001; 2002).

8. WEB USAGE MINING

Most of the previous approaches are concerned with the analysis of the contents of web documents (content mining) or the graph structure of the web (structure mining). Additional information can be inferred from data sources that capture the interaction of users with a web site, e.g., from server-side web logs or from client-side applets that observe a single user's browsing patterns. Such information may, e.g., provide important clues for restructuring web sites (Perkowitz & Etzioni, 2000; Berendt, 2002), personalizing web services (Mobasher, Cooley & Srivastava, 2000; Mobasher, Dai, Luo & Nakagawa, 2002; Pierrakos, Paliouras, Papatheodorou & Spyropoulos, 2003), optimizing search engines (Joachims, 2002), recognizing web spiders (Tan & Kumar, 2002) and many more. An excellent overview and taxonomy of this research area can be found in (Srivastava et al., 2000).

As an example, let us consider systems that make user-specific browsing recommendations (Armstrong et al., 1995; Pazzani et al., 1996; Balabanovic,´ & Shoham, 1995). For example, the WebWatcher system (Armstrong et al., 1995) predicts which links on the currently viewed page are most interesting to the user's search goal, which has to be specified in advance, and recommends the user to follow these links. However, these early systems rely on user intervention by specification of a search goal (Armstrong et al., 1995) or explicit feedback about interesting or not interesting pages (Pazzani et al., 1996). More advanced systems try to infer this information from web logs, thereby removing the need for user feedback. For example, Personal WebWatcher (Mladenić, 1996) is an early attempt that replaces WebWatcher's requirement for an explicitly specified search goal with a user model that has been inferred by a text classification system trained on pages that the user has been observed to visit (positive examples) or not to visit (negative examples). These pages have been obtained by a client-side applet that logs the user's browsing behavior.

More recently, it was tried to infer this information from server-side web logs (Mobasher et al., 2000). The information contained in a web log includes the IP-address of the client, the page that has been retrieved, the time at which the request was initiated, the page from which the link originated, the browsing agent used, etc. However, unless additional information is used (e.g., session cookies), there is no way to reliably determine the browsing path that a user takes. Problems include missing page requests because of client-side caches or merged sessions because of multiple users operating from the same IP-addresses. Special techniques have to be used to infer the browsing paths (so-called *click streams*) of individual users (Cooley, Mobasher & Srivastava, 1999). These click-streams can then be mined using clustering and association rule finding techniques, and the resulting models be used for making page recommendations. The WUM Web Utilization Miner (Spiliopoulou, 1999) is a publicly available, prototypical system that allows to mine web logs using advanced association rule discovery algorithms.

9. COLLABORATIVE FILTERING

Collaborative filtering (Goldberg, Nichols, Oki & Terry, 1992) may be considered a special case of usage mining, which relies on previous recommendations by other users in order to predict which among a set of items are most interesting for the current user. Such systems are also known as *recommender systems* (Resnick & Varian, 1997). Naturally, recommender systems have many applications, most notably in E-commerce (Schafer, Konstan & Riedl, 2000), but also in science (e.g., assigning papers to reviewers; Basu, Hirsh, Cohen & Nevill-Manning, 2001).

Recommender systems typically store a data table that records for each user/item pair whether the user made a recommendation for the item or not and possibly also the strength of this recommendation. Such recommendations can either be made explicitly by giving some sort of feedback (e.g., by assigning a rating to a movie) or implicitly (e.g., by buying a video of the movie). The elegant idea of collaborative filtering systems is that recommendations can be based on user similarity, and that user similarity can in turn be defined by the similarity of their recommendations. Alternatively, recommender systems can also be based on item similarities, which are defined via the recommendations of the users that recommended the items in question (Sarwar, Karypis, Konstan & Riedl, 2001).

Early recommender systems followed an *memory-based* approach, which means that they directly computed this similarity for each new query. For example, the GroupLens system (Konstan, Miller, Maltz, Herlocker, Gordon & Riedl, 1997) required readers of Usenet news articles to rate an article on a scale with five values. From that, similarities between users are cached by computing a correlation coefficient over their votes for individual items.

In a landmark paper, Breese, Heckerman, and Kadie (1998) compare memory-based approaches to *model-based* approaches, which use the stored data for inducing an explicit model for the recommendations of the users. The results show that a Bayesian network outperforms alternative approaches, in particular memory-based approaches. Other types of models that have been studied include clustering (Ungar and Foster, 1998), latent semantic models (Hofmann & Puzicha, 1999) and association rules (Lin, Alvarez & Ruiz, 2002).

An active research area is to combine integrate collaborative filtering with content-based approaches to recommender systems, i.e., approaches that make predictions based on background knowledge of characteristics of users and/or items. An interesting approach is followed by Cohen and Fan (2000), where the use of artificial users are proposed. These users are models of user groups or item groups that can be learned by content-based analysis techniques. For example, an artificial user could represent a certain musical genre and comment positively on all representatives of that genre. Melville, Mooney, and Nagarajan (2002) propose a similar approach by suggesting the use of content-based predictions for replacing missing recommendations. Popescul, Ungar, Pennock, and Lawrence (2001) extend the approach taken by (Hofmann & Puzicha, 1999), which associates users and items with a hidden layer of emerging concepts, by merging word occurrence information into the latent models.

10. CONCLUSION

Web mining is a very active research area. A survey like this can only scratch on the surface. We tried to include references to the most important works in this areas, but we necessarily had to be selective. Nevertheless, we hope to have provided the reader with a good starting point for her own explorations into this rapidly expanding and exciting research area.

11. REFERENCES

1. R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the world-wide web. *Nature*, 401:130–131, September 1999.
2. I. Androutsopoulos, G. Paliouras, and E. Michelakis. Learning to filter unsolicited commercial e-mail. Technical Report 2004/2, NCSR Demokritos, March 2004.
3. R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. WebWatcher: A learning apprentice for the world wide web. In C. Knoblock and A. Levy, editors,

Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, pages 6–12. AAAI Press, 1995. Technical Report SS-95-08.

4. M. Balabanovic,´ and Y. Shoham. Learning information retrieval agents: Experiments with automated web browsing. In C. Knoblock and A. Levy, editors, *Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pages 13–18. AAAI Press, 1995. Technical Report SS-95-08.
5. C. Basu, H. Hirsh, W. W. Cohen, and C. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research*, 14: 231–252, 2001.
6. B. Berendt. Using site semantics to analyze, visualize, and support navigation. *Data Mining and Knowledge Discovery*, 6(1): 37–59, 2002.
7. B. Berendt, A. Hotho, and G. Stumme. Towards semantic web mining. In I. Horrocks and J. Hendler, editors, *Proceedings of the 1st International Semantic Web Conference (ISWC-02)*, pages 264–278. Springer-Verlag, 2002.
8. T. Berners-Lee, R. Cailliau, A. Loutonen, H. Nielsen, and A. Secret. The World-Wide Web. *Communications of the ACM*, 37(8):76–82, 1994.
9. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.
10. 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.
11. 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.
12. 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.
13. J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, Madison, WI, 1998. Morgan Kaufmann.
14. 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.
15. 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).
16. R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Scott Schoenberg. Frequently-asked question files: Experiences with the FAQ finder system. *AI Magazine*, 18(2):57–66, 1997.
17. R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *Proceedings of 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 462–468. AAAI Press, 1996.

18. M. E. Califf, editor. Machine Learning for Information Extraction: Proceedings of the AAAI-99 Workshop, 1999. AAAI Press. Technical Report WS-99-11.
19. M. E. Califf. Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research*, 4:177–210, 2003.
20. S. Chakrabarti. Data mining for hypertext: A tutorial survey. *SIGKDD explorations*, 1(2):1–11, January 2000.
21. S. Chakrabarti. Mining the Web: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann, 2002.
22. 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.
23. 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.
24. 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.
25. W. W. Cohen. Learning rules that classify e-mail. In M. Hearst and H. Hirsh, editors, *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pages 18–25. AAAI Press, 1996. Technical Report SS-96-05.
26. W. W. Cohen and W. Fan. Web-collaborative filtering: Recommending music by crawling the web. In *Proceedings of the 9th International World Wide Web Conference (WWW-9)*, 2000.
27. R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1): 5–32, 1999.
28. 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
29. M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1-2):97–119, 2001.
30. 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.
31. E. Crawford, J. Kay, and E. McCreath. IEMS – the intelligent email sorter. In C. Sammut and A. G. Hoffmann, editors, *Proceedings of the 19th International Conference on Machine Learning (ICML-02)*, pages 263–272, Sydney, Australia, 2002. Morgan Kaufmann.
32. 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.
33. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
34. 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.
35. A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Y. Halevy. Learning to match ontologies. *VLDB Journal*, 12(4):303–319, 2003. Special Issue on the Semantic Web.

36. R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison-shopping agent for the World-Wide Web. In *Proceedings of the 1st International Conference on Autonomous Agents*, pages 39–48, Marina del Rey, CA, 1997.
37. S. Džeroski and N. Lavrač, editors. Relational Data Mining: Inductive Logic Programming for Knowledge Discovery in Databases. Springer-Verlag, 2001.
38. L. Eikvil. Information extraction from world wide web – a survey. Technical Report 945, Norwegian Computing Center, 1999-
39. O. Etzioni. Moving up the information food chain: Deploying softbots on the world wide web. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1322–1326. AAAI Press, 1996.
40. O. Etzioni and D. Weld. A softbot-based interface to the internet. *Communications of the ACM*, 37(7):72–76, July 1994. Special Issue on *Intelligent Agents*.
41. 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.
42. T. Fawcett. "In vivo" spam filtering: A challenge problem for data mining. *SIGKDD explorations*, 5(2), December 2003.
43. D. Fensel. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2001.
44. D. Freitag. Information extraction from HTML: Application of a general machine learning approach. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*. AAAI Press, 1998.
45. J. Fürnkranz. A study using *n*-gram features for text categorization. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence, Wien, Austria, 1998.
46. J. Fürnkranz. Hyperlink ensembles: A case study in hypertext classification. *Information Fusion*, 3(4):299–312, December 2002. Special Issue on Fusion of Multiple Classifiers.
47. J. Fürnkranz, C. Holzbaur, and R. Temel. User profiling for the Melvil knowledge retrieval system. *Applied Artificial Intelligence*, 16(4): 243–281, 2002.
48. 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.
49. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave and information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.
50. G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. The Lixto data extraction project — back and forth between theory and practice. In *Proceedings of the Symposium on Principles of Database Systems (PODS-04)*, 2004.
51. P. Graham. Better bayesian filtering. In *Proceedings of the 2003 Spam Conference (*`http://spamconference.org/proceedings2003.html`*)*, Cambridge, MA, 2003
52. G. Grieser, K. P. Jantke, S. Lange, and B. Thomas. A unifying approach to HTML wrapper representation and learning. In S. Arikawa and S. Morishita, editors, *Proc. 3rd International Conference on Discovery Science*, number 1967 in Lecture Notes in Artificial Intelligence, pages 50–64. Springer–Verlag, 2000.

53. T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 688–693, 1999
54. C.-N. Hsu and M. T. Dung. Generating finite-state transducers for semistructured data extraction from the web. *Information Systems*, 23(8):521–538, 1998. Special Issue on Semistructured Data.
55. T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 133–142. ACM Press, 2002.
56. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999. ISSN 0004-5411.
57. J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997. Special Issue on Recommender Systems.
58. R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD explorations*, 2(1):1–15, 2000
59. R. Kozierok and P. Maes. Learning interface agents. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)*, pages 459–465. AAAI Press, 1993.
60. N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118:15–68, 2000.
61. K. Lang. NewsWeeder: Learning to filter netnews. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ML-95)*, pages 331–339. Morgan Kaufmann, 1995.
62. Y. Lashkari, M. Metral, and P. Maes. Collaborative interface agents. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 444–450, Seattle, WA, 1994. AAAI Press.
63. S. Lawrence and C. L. Giles. Searching the world wide web. *Science*, 280:98–100, 1998.
64. K. Lerman, S. N. Minton, and C. A. Knoblock. Wrapper maintenance: A machine learning approach. *Journal of Artificial Intelligence Research*, 18: 149–181, 2003.
65. M. Levene, J. Borges, and G. Louizou. Zipf's law for Web surfers. *Knowledge and Information Systems*, 3(1): 120–129, 2001.
66. D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Devlopment in Information Retrieval*, pages 37–50, 1992.
67. W. Lin, S. A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1): 83–105, 2002.
68. A. Maedche, C. Nédellec, S. Staab, and E. Hovy, editors. *Proceedings of the 2nd Workshop on Ontology Learning (OL-2001)*, volume 38 of *CEUR Workshop Proceedings*, Seattle, WA, 2001. IJCAI-01.
69. A. Maedche, V. Pekar, and S. Staab. Ontology learning part one — on discovering taxonomic relations from the web. In N.Zhong, J. Liu, and Y. Y. Yao, editors, *Web Intelligence*, pages 301–321. Springer-Verlag, 2003.
70. A. Maedche and S. Staab. Learning ontologies for the semantic web. *IEEE Intelligent Systems*, 16(2), 2001.
71. P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40, July 1994. Special Issue on *Intelligent Agents*.

72. 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.
73. A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In M. Sahami, editor, *Learning for Text Categorization: Proceedings of the 1998 AAAI/ICML Workshop*, pages 41–48, Madison, WI, 1998. AAAI Press.
74. P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002)*, pages 187–192, Edmonton, Canada, 2002.
75. D. Mladenić. Personal WebWatcher: Implementation and design. Technical Report IJS-DP-7472, Department of Intelligent Systems, Joz, ef Stefan Institute, 1996.
76. D. Mladenić. Feature subset selection in text-learning. In C. Nédellec and C. Rouveirol, editors, *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, pages 95–100, Chemnitz, Germany, 1998a. Springer-Verlag.
77. D. Mladenić. Turning Yahoo into an automatic web-page classifier. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 473–474, Brighton, U.K., 1998b. Wiley.
78. D. Mladenić. Text-learning and related intelligent agents: A survey. *IEEE Intelligent Systems*, 14(4):44–54, July/August 1999.
79. D. Mladenić and M. Grobelnik. Word sequences as features in text learning. In *Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK-98)*, Ljubljana, Slovenia, 1998. IEEE section.
80. B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.
81. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6(1): 61–82, 2002.
82. K. J. Mock. Hybrid hill-climbing and knowledge-based methods for intelligent news filtering. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 48–53. AAAI Press, 1996.
83. J. Myllymaki. Effective web data extraction with standard XML technologies (HTML). In *Proceedings of the 10th International World Wide Web Conference (WWW-01)*, Hong Kong, May 2001.
84. 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.
85. T. R. Payne and P. Edwards. Interface agents that learn: An investigation of learning issues in a mail agent interface. *Applied Artificial Intelligence*, 11(1): 1–32, 1997.
86. M. T. Pazienza, editor. Information Extraction in the Web Era: Natural Language Communication for Knowledge Acquisition and Intelligent Information Agents (SCIE-02), Rome, Italy, 2003. Springer-Verlag.
87. M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 54–61. AAAI Press, 1996.
88. M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118:245–275, 2000.

89. D. Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction*, 13 (4):311–372, 2003.
90. A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 437–444. Morgan Kaufmann, 2001.
91. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
92. 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.
93. P. Resnick and H. R. Varian. Special issue on recommender systems. *Communications of the ACM*, 40(3), 1997.
94. 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.
95. E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049. AAAI Press, 1996a.
96. E. Riloff. An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, 85:101–134, 1996b.
97. G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading, MA, 1989.
98. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24 (5):513–523, 1988.
99. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, November 1975.
100. B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference (WWW-10)*, Hong Kong, May 2001.
101. J. B. Schafer, J. A. Konstan, and J. Riedl. Electronic commerce recommender applications. *Data Mining and Knowledge Discovery*, 5(1/2): 115–152, 2000.
102. T. Scheffer. Email answering assistance by semi-supervised text classification. *Intelligent Data Analysis*, 8(5), 2004.
103. S. Scott and S. Matwin. Feature engineering for text classification. In I. Bratko and S. Džeroski, editors, *Proceedings of 16th International Conference on Machine Learning (ICML-99)*, pages 379–388, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.
104. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.
105. B. Sheth and P. Maes. Evolving agents for personalized information filtering. In *Proceedings of the 9th Conference on Artificial Intelligence for Applications (CAIA-93)*, pages 345–352. IEEE Press, 1993.
106. 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.
107. S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1–3):233–272, 1999.

108. 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).
109. M. Spiliopoulou. The laborious way from data mining to web log mining. *Journal of Computer Systems Science and Engineering*, 14:113–126, 1999. Special Issue on Semantics of the Web.
110. 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.
111. S. Staab and A. Maedche. Knowledge portals — ontologies at work. *AI Magazine*, 21(2):63–75, Summer 2001.
112. S. Staab, A. Maedche, C. Nédellec, and P. Wiemer-Hastings, editors. *Proceedings of the 1st Workshop on Ontology Learning (OL-2000)*, volume 31 of *CEUR Workshop Proceedings*, Berlin, 2000. ECAI-00.
113. G. Stumme, A. Hotho, and B. Berendt, editors. *Proceedings of the ECML/PKDD-01 Workshop on Semantic Web Mining*, Freiburg, Germany, 2001.
114. G. Stumme, A. Hotho, and B. Berendt, editors. *Proceedings of the ECML/PKDD-02 Workshop on Semantic Web Mining*, Helsinki, Finland, 2002.
115. P.-N. Tan and V. Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, 6(1): 9–35, 2002.
116. L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. In H. Kautz, editor, *Proceedings of the AAAI-98 Workshop on Recommender Systems*, page 112, Madison, Wisconsin, 1998. AAAI Press. Technical Report WS-98-08.
117. Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. Fisher, editor, *Proceedings of the 14th International Conference on Machine Learning (ICML-97)*, pages 412–420, Nashville, TN, 1997. Morgan Kaufmann.
118. 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.