

Finding, Extracting and Exploiting Structure in Text and Hypertext

Ola Ågren



PHD THESIS, 2009
DEPARTMENT OF COMPUTING SCIENCE
UMEÅ UNIVERSITY

COPYRIGHT © OLA ÅGREN 2009

EXCEPT PAPER:

I COPYRIGHT © OLA ÅGREN 2001

II COPYRIGHT © CSREA PRESS 2002

III COPYRIGHT © KNOWLEDGE SYSTEMS INSTITUTE 2003

IV COPYRIGHT © CSREA PRESS 2003

V COPYRIGHT © OLA ÅGREN 2006

VI COPYRIGHT © EMERALD GROUP PUBLISHING 2008

UMINF 09.12

ISSN 0348-0542

ISBN 978-91-7264-799-2

To my family

Till min familj

Abstract

Data mining is a fast-developing field of study, using computations to either predict or describe large amounts of data. The increase in data produced each year goes hand in hand with this, requiring algorithms that are more and more efficient in order to find interesting information within a given time.

In this thesis, we study methods for extracting information from semi-structured data, for finding structure within large sets of discrete data, and to efficiently rank web pages in a topic-sensitive way.

The information extraction research focuses on support for keeping both documentation and source code up to date at the same time. Our approach to this problem is to embed parts of the documentation within strategic comments of the source code and then extracting them by using a specific tool.

The structures that our structure mining algorithms are able to find among crisp data (such as keywords) is in the form of subsumptions, i.e. one keyword is a more general form of the other. We can use these subsumptions to build larger structures in the form of hierarchies or lattices, since subsumptions are transitive. Our tool has been used mainly as input to data mining systems and for visualisation of data-sets.

The main part of the research has been on ranking web pages in a such a way that both the link structure between pages and also the content of each page matters. We have created a number of algorithms and compared them to other algorithms in use today. Our focus in these comparisons have been on convergence rate, algorithm stability and how relevant the answer sets from the algorithms are according to real-world users.

The research has focused on the development of efficient algorithms for gathering and handling large data-sets of discrete and textual data. A proposed system of tools is described, all operating on a common database containing “fingerprints” and meta-data about items. This data could be searched by various algorithms to increase its usefulness or to find the real data more efficiently.

All of the methods described handle data in a crisp manner, i.e. a word or a hyper-link either is or is not a part of a record or web page. This means that we can model their existence in a very efficient way. The methods and algorithms that we describe all make use of this fact.

Keywords

Automatic propagation; CHiC; data mining; discrete data; extraction; hierarchies; ProT; rank distribution; S²ProT; spatial linking; web mining; web searching

Sammanfattning

Informationsutvinning (som ofta kallas data mining även på svenska) är ett forskningsområde som hela tiden utvecklas. Det handlar om att använda datorer för att hitta mönster i stora mängder data, alternativt förutsäga framtida data utifrån redan tillgänglig data. Eftersom det samtidigt produceras mer och mer data varje år ställer detta högre och högre krav på effektiviteten hos de algoritmer som används för att hitta eller använda informationen inom rimlig tid.

Denna avhandling handlar om att extrahera information från semi-strukturerad data, att hitta strukturer i stora diskreta datamängder och att på ett effektivt sätt rangordna webbsidor utifrån ett ämnesbaserat perspektiv.

Den informationsextraktion som beskrivs handlar om stöd för att hålla både dokumentationen och källkoden uppdaterad samtidigt. Vår lösning på detta problem är att låta delar av dokumentationen (främst algoritmbeskrivningen) ligga som blockkommentarer i källkoden och extrahera dessa automatiskt med ett verktyg.

De strukturer som hittas av våra algoritmer för strukturextraktion är i form av underordnanden, exempelvis att ett visst nyckelord är mer generellt än ett annat. Dessa samband kan utnyttjas för att skapa större strukturer i form av hierarkier eller riktade grafer, eftersom underordnandena är transitiva. Det verktyg som vi har tagit fram har främst använts för att skapa indata till ett informationsutvinningssystem samt för att kunna visualisera indatan.

Huvuddelen av den forskning som beskrivs i denna avhandling har dock handlat om att kunna rangordna webbsidor utifrån både deras innehåll och länkarna som finns mellan dem. Vi har skapat ett antal algoritmer och visat hur de beter sig i jämförelse med andra algoritmer som används idag. Dessa jämförelser har huvudsakligen handlat om konvergenshastighet, algoritmernas stabilitet givet osäker data och slutligen hur relevant algoritmernas svarsmängder har ansetts vara utifrån användarnas perspektiv.

Forskningen har varit inriktad på effektiva algoritmer för att hämta in och hantera stora datamängder med diskreta eller textbaserade data. I avhandlingen presenterar vi även ett förslag till ett system av verktyg som arbetar tillsammans på en databas bestående av "fingeravtryck" och annan meta-data om de saker som indexerats i databasen. Denna data kan sedan användas av diverse algoritmer för att utöka värdet hos det som finns i databasen eller för att effektivt kunna hitta rätt information.

Preface

The thesis consists of the six papers listed below and an introductory part. In the introductory part, a general background on data mining is presented, as well as more in depth coverage of the areas that are more closely related to our research. The main findings of our research are described, as well as a proposed system for handling large amounts of discrete and semi-structured data. The main parts of this thesis are followed by an appendix containing a Users' Guide for CHiC (see Paper III).

List of Papers

- I ÅGREN, O. ALGEXT — *an ALGORITHM EXTRACTOR for C Programs*, Technical Report UMINF 01.11, Department of Computing Science, Umeå University, 2001.
- II ÅGREN, O. Automatic Generation of Concept Hierarchies for a Discrete Data Mining System, in *Proceedings of the International Conference on Information and Knowledge Engineering (IKE '02)* (Las Vegas, Nevada, USA, June 24–27, 2002), pp. 287–293.
- III ÅGREN, O. CHiC: A Fast Concept Hierarchy Constructor for Discrete or Mixed Mode Databases, in *Proceedings of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)* (San Francisco, California, USA, July 1–3, 2003), pp. 250–258.
- IV ÅGREN, O. Propagation of Meta Data over the World Wide Web, in *Proceedings of the International Conference on Internet Computing (IC '03)* (Las Vegas, Nevada, USA, June 23–26, 2003), pp. 670–676.
- V ÅGREN, O. Assessment of WWW-Based Ranking Systems for Smaller Web Sites, *INFOCOMP Journal of Computer Science* vol. 5, no. 2 (June 2006), pp. 45–55.
- VI ÅGREN, O. S²ProT: Rank Allocation by Superpositioned Propagation of Topic-Relevance, *International Journal of Web Information Systems* vol. 4, no. 4 (2008), pp. 416–440.

Other Publications

Outside the thesis work, and in addition to the papers listed above, Ola Ågren has (co-)authored the following publications:

- ÅGREN, O. Teaching Computer Concepts Using Virtual Machines, *SIGCSE Bulletin* vol. 31, no. 2 (June 1999), pp. 84–85.
- ÅGREN, O. *The DARK-Series of Virtual Machines*, Technical Report UMINF 00.15, Department of Computing Science, Umeå University, 2000.
- ÅGREN, O. Virtual Machines as an Aid in Teaching Computer Concepts, *IEEE TCCA Newsletter* (September 2000), pp. 72–76.
- ÅGREN, O. *BitSet: Implementing Sets of Natural Numbers Using Packed Bits*, Technical Report UMINF 02.10, Department of Computing Science, Umeå University, 2002.
- BÖRSTLER, J., JOHANSSON, O., LARYD, A., ORCI, T., SEGERBERG, H., AND ÅGREN, O. *Quality Management for Small Enterprises*, Technical Report UMINF 02.20, Department of Computing Science, Umeå University, 2002.
- Editor for the proceedings of Umeå’s student workshop in Computer Architecture, 2000–2006.

Acknowledgements

A thesis is not something that can be done in isolation and there has been a lot of input from various sources that I am extremely grateful for.

My thesis supervisor, Associate Professor Jürgen Börstler, for giving me more or less free hands to pursue my own personal interests, for commenting on the almost endless sets of drafts, and long discussions on various parts of what research is and Computing Science (especially Software Engineering).

My thesis co-supervisor, Associate Professor Frank Drewes, for discussing the more technical aspects of what I have been working on, for creative revisions, and for being there as a friend and a former (and future?) Table Tennis team mate.

The staff at the Department of Computing Science, especially Steven Hegner, Michael Minock, Lena Kallin Westin, Per-Åke Wedin, and *all* of the administration and support staff. This includes all those that inspired me but have moved on in life: Peter Jacobson, Olof Johansson, Krister Dackland, and those that have worked as teaching assistants in my courses over the years.

The staff at the Department of Interactive Media and Learning, you made me feel welcome and gave me support.

On a more personal level I must say that I'm unable to thank my friends enough. You already know who you are, but a short list of your names¹ includes: Anders, Anna, Anne, Annelie, Annika, Anton, Bertil, Björn, Britta, Cecilia, Claes, Clas, Claudia, Daniel, David, Elin, Elina, Emelie, Emilott, Eric, Erik, Erika, Eva, Fredrik, Frida, Gunnar, Göran, Hanna, Hans, Helena, Henrik, Ingemar, Ingrid, Jan, Jannek, Jenni, Jennie, Jennifer, Jenny, Jens, Jeroen, Joakim, Johan, Johanna, Jonas, Jörgen, Katarina, Klas, Krister, Kristina, Lars, Leif, LenaMaria, Lennart, Lina, Linda, Linus, Lisa, Lotta, Lovisa, Magnus, Malin, Marcus, Maria, Marianne, Martin, Mattias, Melker, Mikael, Mona, Mårten, Niclas, Niklas, Nikoletta, Nils, Nina, Ola, Olov, Oskar, Palle, Per, Per-Olof, Peter, Petter, Pär Anders, Rickard, Rikard, Robert, Roger, Runa, Sabina, Sandra, Sara, Sigrid, Simon, Sofi, Stefan, Stephan, Teresa, Therese, Thomas, Tomas, Tommy, Ulrika, Valentin, Viktoria, Viveka, Wenche, Åke, Örjan, and probably some more that I missed. You are the best!

¹Name appears once, even if I know more than one with that name.

Finally, and most importantly, I thank my family; My mother Solveig, father Sten and former wife Anneli for supporting me and allowing me to follow my own paths in life. My son Simon for being the sunshine of my life. My brother Bo, his wife Maria and their fantastic daughters Sanna and Emma for being there for me. My cousins and their families for being great sources of inspiration.

Live Long and Prosper!

Contents

1	Introduction	1
1.1	Research Questions	3
2	Data Mining	5
2.1	Information Extraction	7
2.2	Clustering	7
2.3	Mining for Association Rules	7
2.4	Thesis Contributions	12
3	Web Search Engines	15
3.1	Web Mining	16
3.2	Web Link Mining	17
3.3	Thesis Contributions	22
3.4	Summary	26
4	Final Remarks	29
5	Bibliography	31

Paper I	41
6 ALGEXT — an ALGORITHM EXTRACTOR for C Programs	43
6.1 Introduction	45
6.2 Contents of a C File	46
6.3 Source Code Requirements	48
6.4 Implementation	48
6.5 Examples	49
6.6 Discussion	51
6.7 References	52
6.A Users' Guide	53
6.B System Documentation	54
6.C Comment Comparison vis-à-vis ALGEXT	57
Paper II	59
7 Automatic Generation of Concept Hierarchies for...	61
7.1 Introduction	62
7.2 Definitions	64
7.3 The Algorithm	65
7.4 Example of Execution	68
7.5 Algorithm Analysis	71
7.6 Related Work	72
7.7 Discussion	73
7.8 References	75
7.A All Results from Step 1	76

Paper III	77
8 CHiC: A Fast Concept Hierarchy Constructor for...	79
8.1 Introduction	80
8.2 Background	81
8.3 Definitions	83
8.4 The Algorithm	84
8.5 Example of Execution	92
8.6 Algorithm Analysis	96
8.7 Related Work	99
8.8 Discussion	100
8.9 Experiences	101
8.10 References	102
Paper IV	103
9 Propagation of Meta Data over the World Wide Web	105
9.1 Introduction	106
9.2 Propagation Algorithm	107
9.3 Definitions	110
9.4 A Monotone Data Flow System on Meta Data	111
9.5 Related Work	114
9.6 Discussion	114
9.7 References	115
Paper V	117
10 Assessment of WWW-Based Ranking Systems for Smaller Web Sites	119
10.1 Introduction	120
10.2 Methods and Materials	124
10.3 Results	128
10.4 Discussion and Conclusions	132
10.5 Acknowledgements	134
10.6 References	135
10.A Test Database	136
10.B Keywords	136
10.C Confidence Intervals per Keyword	137
10.D Kolmogorov-Smirnov Results	137

Paper VI	141
11 S²ProT: Rank Allocation by Superpositioned Propagation of Topic-Relevance	143
11.1 Introduction	145
11.2 Preliminaries	147
11.3 Related Works	150
11.4 Propagation of Topic-relevance	152
11.5 Comparison of Algorithm Behaviours	160
11.6 Empirical Results	166
11.7 Discussion	177
11.8 References	179
11.A Theorems and Proofs	181
11.B Search Terms for the Assessment	184
Appendices	185
A Users' Guide to CHIC	187
A.1 Introduction	188
A.2 Using the words Program	188
A.3 Using the chic Program	190
A.4 Contents of Each Data Base File	192
A.5 An Example Data Base	192
A.6 References	196
Index	197
Colophon	201

Chapter 1

Introduction

Each year more and more data is generated in various forms, currently in the multiple Exabytes per year range. Most of this data is in some type of raw format that must be refined before it can be used and understood. Moreover, a lot of this data is stored on magnetic media of some sort for later retrieval.

To give some sense of scale, we will look at the Internet as an example of how much information is available. Figure 1.1 on the following page displays the number of hosts (registered host names) available on the Internet over the last 25 years. Moreover, the figure shows that the data traffic over the Internet has increased even faster. In fact, the sheer volume of data added each week is so large that no one would be able to read all the information within a lifetime. This means that tools must be used to help find interesting information, or even go so far as to draw conclusions from the available data.

The tools might be simple or complex, but they can only work with the data they are given. The simplest tools available can only search for records containing exact matches of the keywords given in the query. Having more information about each document means that we can use more complex tools for that data-set. Two very important attributes here are how structured the data-set is and whether there is additional meta-data for each record.

If the data exist in a database or other forms of formally defined data-set, usually called *structured data*, it can easily be searched and used by software. The so called Deep Web is built up of rich information and databases accessible using forms or other software, and is usually seen as being much larger than the static web [100].

A bigger problem exists if the data has no apparent structure or has only a minimal inherent structure. General text files tend to have no structure outside of those on the syntactical level, and are often seen as a type of *unstructured data*.

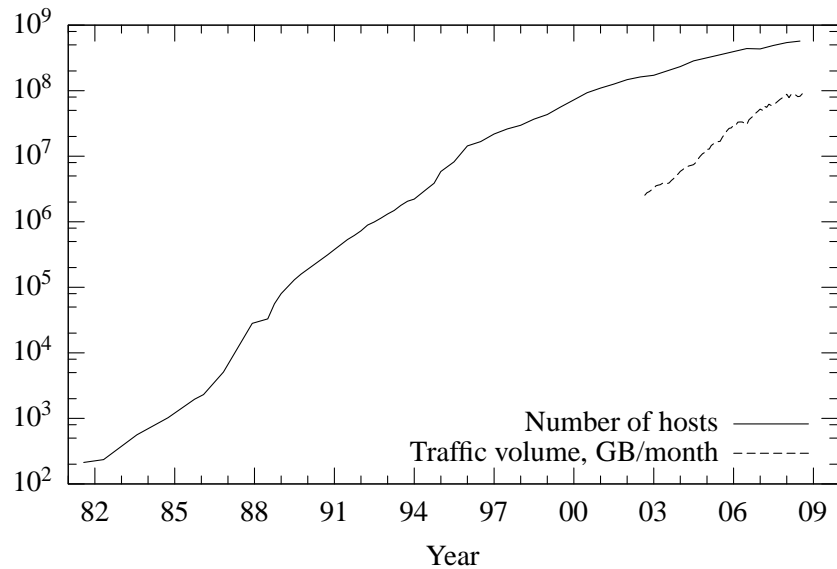


Figure 1.1: The number of hosts on the Internet 1982–2008 [73] and traffic through AMS-IX (the largest Internet exchange point) during 2002–2008 [9].

Some form of natural language processing is usually required to use such data-sets efficiently as soon as something more complex than a pure keyword search has to be done.

There is also the middle ground between structured and unstructured data called *semi-structured data*. A lot of the file types currently found on the Internet (such as HTML and PDF) allow a number of structural parts such as headings and hyperlinks pointing to other documents. This is the type of data that we have focused on, even though our indexing works on unstructured data as well.

It is also possible to use descriptive information about the data rather than the actual contents of the data. Such information is called *meta-data* and usually contains information about elements and attributes, records and structures, and provenance of the data. Typical examples of meta-data include the name, size, data type and length of each field available in the data-set, as well as where it is located, its association to other data, ownership of the data, etc. Meta-data can sometimes be seen as a model of the original data, thereby allowing applications and users to search and browse the available meta-data rather than the original data-set. As an example, the abstract of a book is together with the CIP record¹ supposed to give a

¹A Cataloging in Publication record is a bibliographic record prepared by the American Library of Congress for a book that has not yet been published. When the book is published, the publisher

selling but objective sampling of the content of a book without going into details. This can be seen as a case where meta-data is used to enhance the usefulness of the data it describes [101]. An analogue would be to use thumbnails to provide an overview of all available pictures in a gallery rather than using the full pictures directly.

This thesis collects the results from three different projects dealing with semi-structured data or meta-data. The first one, called *ALGEXT*, extracts meta-data from source code. The second one, called *CHIC*, finds structure within large sets of discrete meta-data. The last, and most important, one is the *PROT* project. It uses structural as well as textual elements from semi-structured documents in order to rank them, and is by far the most complex of the three projects.

1.1 Research Questions

The main questions that the research in this thesis tries to answer are:

- How can we find and extract structural information or embedded data from discrete data-sets?
- How can we find and rank web-pages in a topic-sensitive way by algorithms that are more efficient (at least in practice) than the currently known ones?

includes the CIP data on the copyright page. This makes book processing easier for both libraries and book dealers [8].

Chapter 2

Data Mining

Data mining is a collective term that stands for a number of different procedures and methods for finding interesting information in large amounts of data. Other names used for the concept include deductive learning, exploratory data analysis, and data driven discovery [47].

While data mining can be applied to any type of data-set, it has been used extensively in business systems. Data mining tools usually do not work directly on a “live” database that contains day to day transactions, but operate on a modified and summarised database called a *data warehouse*. A data warehouse contains aggregated and cleaned information from the “live” databases, i.e. ideally no false or extraneous data [47, 71, 133].

Another difference between a regular database system and a data mining system is in their operation. The user of a database system expects a crisp answer to each query, for example, is a seat available on a certain flight. The answer given by a data mining system could be in the form of possibly interesting patterns or meta-data describing something in the database, for example, every user that bought article *a* also bought article *b* [47].

Data mining approaches are usually grouped into either predictive or descriptive systems, according to the taxonomy in Figure 2.1.

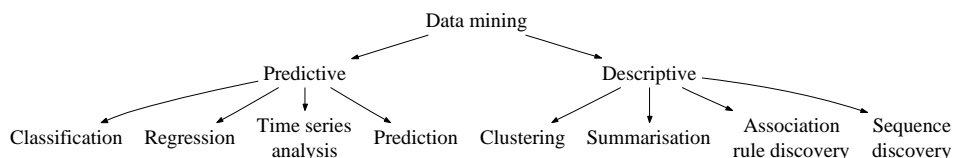


Figure 2.1: Data mining taxonomy [47].

A *predictive* system makes some sort of a prediction for new values based on already known values in the data-set. Predictive approaches include classification, regression, time series analysis and prediction systems.

- A *classification* system maps each object into one of a number of predefined classes.
- A *regression* system finds some sort of function that as closely as possible matches the given data. One of the the attributes of the data is modelled by the other attributes using the function.
- *Time series analysis* examines the behaviour of a value over time.
- A *prediction* system tries to foresee a future state given current and previous states. These systems are often used to give early warning for natural phenomena, like earthquakes and flooding, as well as other semi-unpredictable systems like speech recognition and pattern recognition [47].

A *descriptive* system tries to find relationships in data, e.g. patterns. Most of the time it is not used to predict future values, but can be used to analyse different attributes in the data. Descriptive approaches include clustering, summarisation, association rule discovery and sequence discovery.

- *Clustering* is related to classification, but creates the classes by using the existing data.
- *Summarisation* diminishes the amount of data, while retaining as much significant information as possible about the initial data set.
- *Association rule discovery* tries to find associations between different items in the database.
- *Sequence discovery* looks for patterns where one event leads to another event.

Most of our work has been on different descriptive approaches, including information extraction (see Section 2.1), clustering (see Section 2.2), mining for association rules (see Section 2.3) and web-based data mining (see Chapter 3).

2.1 Information Extraction

Information extraction is the process of extracting, computing and compiling information from the text of a large corpus using machine learning [61]. Information extraction systems are generally used to extract information about a page, storing it in a form that makes queries and retrieval of the data as easy and efficient as possible. Typical examples of information extraction systems include RSV [53] and WHISK/CRYSTAL [125, 126].

2.2 Clustering

One widely used form of descriptive data mining is *clustering*. Clustering is related to classification, but uses the existing data to automatically generate the classes. Clustering is also called *unsupervised learning* or *segmentation*. A very important notion in clustering is *similarity*, i.e. how closely related two (or more) items are to each other. Clustering works by automatically grouping together smaller clusters (i.e. data points with similar values) until either each cluster is “sufficiently” large or a certain (predefined) number of clusters have been reached.

Clustering can be seen as finding groups of facts not previously known in large data. There are numerous ways of clustering data-sets, depending on, for example, type and distribution of data. An excellent review of the state of the art in data clustering was published by Jain et al. in 1999 [76].

2.3 Mining for Association Rules

Mining for association rules is looking for patterns according to which one item is connected to another item. There are many different applications available supporting mining for association rules. Most of them fall into two different categories, unsupervised and supervised mining (see Sections 2.3.1 and 2.3.2, respectively).

The rules found are usually in the form of an implication $X \Rightarrow Y$. Each rule found is marked with the quality attributes called support and confidence. A formal definition is included in order to introduce the notation and terminology used later in this text.

DEFINITION 2.1 Let $I = \{I_1, I_2, \dots, I_m\}$ be a set of *items* or an *itemset*. Let \mathcal{D} be a set of *transactions*, where each transaction T is a set of items, $T \subseteq I$. We say that a transaction T *contains* X if $X \subseteq T$. The fraction of transactions containing X is called the *frequency* of X . An *association rule* is an implication in the form $Y \Rightarrow Z$, where $Y, Z \subseteq I$, and $Y \cap Z = \emptyset$. This rule holds in the transaction set \mathcal{D} with *confidence* α if the fraction of the transactions containing Y that also contain Z is at least α . The rule has *support* s in the transaction set \mathcal{D} if the fraction of the transactions in \mathcal{D} that contain $Y \cup Z$ is at least s [47].

EXAMPLE 2.1 Given a database with three items $I = \{i_1, i_2, i_3\}$ and five transactions $D = \{\{i_1\}, \{i_1, i_2\}, \{i_1, i_2, i_3\}, \{i_2, i_3\}, \{i_2\}\}$, we can say that the support for $i_1 \Rightarrow i_2$ is $s(i_1 \Rightarrow i_2) = \frac{2}{5} = 40\%$ and the confidence is $\alpha(i_1 \Rightarrow i_2) = \frac{2}{3} \approx 67\%$. We can also see that $s(i_2 \Rightarrow i_3) = \frac{2}{5} = 40\%$ and $\alpha(i_2 \Rightarrow i_3) = \frac{2}{4} = 50\%$, so $i_1 \Rightarrow i_2$ has more confidence than $i_2 \Rightarrow i_3$ while they have equal support.

2.3.1 Unsupervised Mining for Association Rules

Unsupervised association rule mining systems automatically search \mathcal{D} to discover association rules having high confidence and support, without being guided by input from the user. The most commonly used algorithm is called *Apriori* [47, 129].

Apriori builds upon the fact that only subsets of large sets can be large, and the assumption that only large subsets can give new information that is potentially important. This means that the possible solution space can be pruned quickly while checking the combination of all itemsets that differ in only one member. A support parameter s is used in Apriori to decide which itemsets are considered large. This means that the algorithm will ignore rules with high confidence if the support is too small [47, 64, 129].

Example 2.2 on the facing page shows how Apriori would prune the solution space using already found information. Unsupervised data mining will not be discussed further in this thesis, since it is not the focus of the work described here.

EXAMPLE 2.2 Suppose we are given a data-set with three items (A , B and C), and two of these (A and B) appear frequently while the last one (C) does not. Combining item C with either A or B results in an itemset that is not frequent enough, thus indicating that these can safely be ignored from further calculations by Apriori. This implies that only the intersection of A and B (denoted $\{A, B\}$) can be a frequent itemset when combining these itemsets. This is illustrated in the lattice in Figure 2.2.

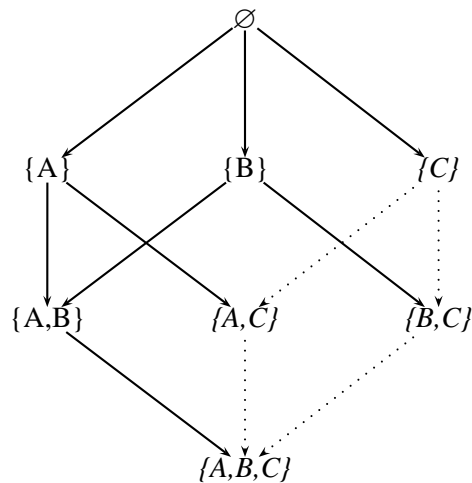


Figure 2.2: Lattice of itemsets for Example 2.2, with frequent and *infrequent* itemsets. Dotted lines can safely be ignored by Apriori, since at least one of its parents are infrequent.

2.3.2 Supervised Mining for Association Rules

Supervised mining for association rules is often performed on a data warehouse, where the available data is already somewhat summarised and cleaned. The data that is operated upon is usually described not only by a value, but also by a number of attributes that describe from whom and where it came [47, 64].

EXAMPLE 2.3 Assume that we have a company that has two branches in Umeå and one in Stockholm (both cities are in Sweden) as well as one in London, England. The company sells electronics (television sets and portable CD players, among other products) and mirror shades. To improve its operations the company wants to use the sales data gathered from each branch to make estimates of the number of units of each product type which need to be preordered. Each branch of the company has sent in the sales figures for each product type for each day to the central data warehouse that houses this data in a database. This means that each data value is marked with a number of attributes, such as date, location, and product type.

Each attribute of Example 2.3 can be seen as a hierarchy on its own, as illustrated in Figure 2.3. Such hierarchies are often referred to as *concept hierarchies* [47, 64]. We have explored automatic generation of concept hierarchies using CHiC [140, 141, Papers II–III].

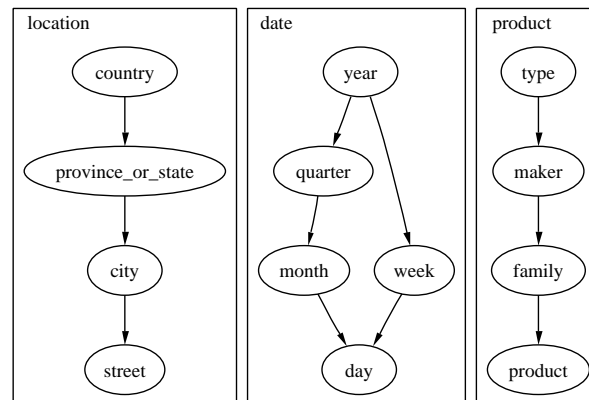


Figure 2.3: Examples of concept hierarchies for Example 2.3.

The usual way of looking at the data in a supervised system is in the form of an n -dimensional *data cube*, such as the one in Figure 2.4. Each side of the data cube corresponds to the current view level in the hierarchy of the attribute, also called *facet*. Another name for a data cube is *On-Line Analytical Processing (OLAP) cube* [47, 64, 129].

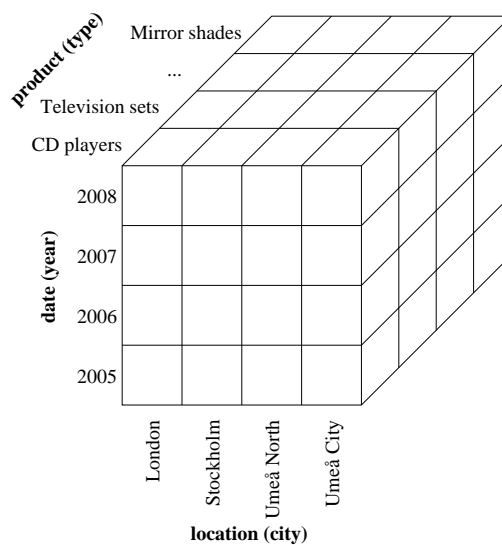


Figure 2.4: Data cube corresponding to Example 2.3 and the hierarchies (facets) shown in Figure 2.3 on the facing page.

Each block in the cube corresponds to the chosen hierarchy level of each facet. It is marked with the aggregated values of all underlying levels in the hierarchy as well as its support (see Definition 2.1 on page 8). This cube can be changed and examined by the user to find interesting patterns.

It is up to the user to choose operations (see Table 2.1 on the following page) in such a way that new information can be deduced from the database. This means that the output from using supervised mining for association rules will depend on the user expertise in both the domain and the tools used.

A typical starting point would be to look at highly aggregated values over either time (e.g. sales data per year) or per product, and then drilling down to find patterns in the data. This would show variations due to season, locale, or product in our example.

Table 2.1: Typical operations that can be performed on a data cube [47, 64].

<i>Operation</i>	<i>Result</i>
Transpose	Changes the positions of facets with respect to the others
Slice/dice	Choose a specific slice in one (or more) dimension
Drill-down	Goes to a lower level in the hierarchy of one facet ^a
Roll-up	The opposite of drill-down

^aThe data cube in Figure 2.4 on the previous page has already been drilled down to the “city” and “type” level in the location and product facets, respectively.

2.4 Thesis Contributions

There are two separate subareas within data mining that have been studied in the present thesis. Each of them will be handled in its own subsection below.

2.4.1 Algorithm Extraction

A lot of research has been done to ensure that documentation of software is as up-to-date as possible, but there are still some open problems. There is usually a semantic gap between the source code and the documentation for several reasons; the source code and the documentation are not always written at the same time, different programs are probably used to edit them, etc. This means that whenever a change is done in one, it needs to be transferred to the other.

Our approach is to extract some parts of the documentation from the software. This means that the documentation and source share the same file, implying that there is an increased likelihood that the documentation will be updated whenever a change is made to the source statements and vice versa; cf. the simplified version of literate programming seen in `c-web` [52].

While truly automatic extraction of algorithms has not yet been mastered, it is at least possible to use comments in order to add to the source code whatever information is required. ALGEXT [139, Paper I] is a proof of concept implementation that extracts all *strategic* comments from ANSI C (see Example 2.4 on the facing page), retaining the indentation of the source code in the extracted comments.

The main idea is to allow a textual description of the algorithms to be embedded within the source code, and extract it when required. This works in a similar way to `cextract`¹, `doxygen` [134] and `Javadoc` [54] to extract (part of) the function comments of the source files, but with less requirements on the comment markup from the tool’s viewpoint; Example 2.4 shows that the comments can be quite

¹Source code available from <http://dev.w3.org/cvsweb/Amaya/tools/cextract-1.7/>.

elaborate because of the requirements from other tools, in this case \LaTeX .

EXAMPLE 2.4 Given the following source code:

```
/* Function f(x) = x*x*x-x */
int f(int x)
{
    /* \begin{equation} \mathcal{Z} \rightarrow \mathcal{Z}, f(x) = x^3-x */
    return x*x*x-x; /* will not work if x is too large */
    /* \label{eq:f} \end{equation} */
}
```

The \LaTeX embedded in the tactical comments of the source code above generates Eq. (2.1).

f:

$$\mathcal{Z} \rightarrow \mathcal{Z}, f(x) = x^3 - x \quad (2.1)$$

2.4.2 Structure Extraction

The concept hierarchies used when doing supervised mining for association rules (see Section 2.3.2) are normally defined at the same time as the database or by using predefined hierarchies, such as Dublin Core [46] or LOM [72]. This will not work that well when the data-set consists of free-text terms or free-text meta-data describing each record. The main problem is that changing the set of records to be included may yield a different set of terms to use as well. This means that there has to be an automated process to find the concept hierarchies given by the terms.

The process finds subsumptions, i.e. terms that exist in a record only if another term exists there as well but not vice versa. These subsumptions are then used to build up hierarchies that can be used either for semantic searches for documents or for doing supervised mining for association rules among the records.

The use of concept hierarchies to increase the number of documents found has been very successful in information retrieval. A larger set of documents can often be found by enriching the queries with terms that subsume the original terms in the hierarchy [30, 120].

At the turn of the century there were no tools available that could generate a concept hierarchy specifically made for supervised mining for association rules. It was possible to use decision tree inducers that could generate binary trees by checking one attribute at a time, using algorithms that were not optimised for crisp data-sets.

This was the main motivation for creating the CHIC tool, that is able to induce a concept hierarchy of terms given keyword based data [140, 141, Papers II–III]. It

was originally designed to work together with a proprietary data mining system on the IRIX platform, but it is easy to adapt the output to most data mining systems.

There have been two upgrades of functionality in the tool from Paper II [140] to Paper III [141], both being driven by an upgrade of the functionality of the data mining system. The first upgrade was to allow generation of concept lattices rather than hierarchies. This means that more than one path to a subsumed keyword may exist in the resulting data-set. The second upgrade was to allow terms to be reused in different facets, as long as there is no overlap between the keywords of the facets. Reusing terms increases the coverage in the later facets. Turning these options on means an increased amount of work required to generate the results (see Paper III [141, Sections 8.6.1 to 8.6.2 on pages 96–98]).

The clustering generated by CHIC is not guaranteed to be optimal, since the algorithm uses local minima to select decision points. Experience shows that it generates appropriate results in almost all practical cases; We tested thousands of data-sets and found only three hierarchies that did not quite make sense.



Figure 2.5: Problems with applying Data Mining on sales data.

With permission from PIB Copenhagen A/S 3/2004.

Chapter 3

Web Search Engines

What is currently known as Internet started out as a research network with packet switched data called ARPANET. It grew larger and larger as more computers and networks were added to it over time, and some of the older protocols were replaced to get more stability and/or throughput.

It was first and foremost used for transmitting text messages and text files, until Tim Berners-Lee from CERN in Switzerland created the first working prototype of what is now known as the World Wide Web. It consisted of a web server, a combined browser and editor, and a number of pages that described the project. Some of the technologies that we now take for granted were first introduced in this project, e.g. globally unique identifiers (Uniform Resource Identifier).

There were originally very few servers up and running, so it was possible to keep track of all of them and then manually browse to find the wanted material. It did, however, not take long until the number of servers was too large to keep track of manually (see Figure 1.1 on page 2). This meant that some sort of look-up service was required.

Along came the first generations of *web search engines* [20, Section 4.72], e.g. AltaVista.¹ They indexed all pages they could reach and provided their users with an easy way of doing searches. They usually had no way of ranking the pages, instead they gave their answers in an unspecified (albeit usually deterministic) order. The key to using these search engines was to add enough search terms (both positive and negative) to a query to get the right number of pages.

Over the years more and more advanced search engines appeared. These search engines used various techniques to give better search results. Among the most successful and prominent ones is the idea to use the links between web pages to

¹Their original search engine became operational in 1995 and was located at <http://www.altavista.com>. They have later created far more advanced search engines.

derive rankings indicating the relative importance of pages. Approaches based on this idea will be discussed in this chapter.

3.1 Web Mining

Web mining is data mining using data from the web. Within this field, there are the following five major research areas:

Information extraction Finding, extracting and compiling information from a large corpus, see Section 2.1 on page 7.

Wrapper induction The process of finding general structural information about a set of web pages, and with this in mind extract only the relevant information from each page [36, 108, 109].

Vector space modelling and (latent) semantic indexing A method for extracting and representing the similarity between documents and the meaning of words from the contexts, by applying statistical computations to a large corpus of text [94, 119, 129].

Web link mining Mining the spatial link structure of the web for information, see Section 3.2.

Web log mining Mining for knowledge in web logs, otherwise known as *click-stream* data [25, 48, 103, 137].

3.2 Web Link Mining

The main part of our work concerns web link mining. A lot of research has been done by exploring the link structure between pages², especially about algorithms for very large data-sets such as the entire world wide web. Pages on a specific subject tend to have links to other pages on the same subject [42, 51, 81]. Neighbouring web pages (when using hyper-links to define distance) can be used to either deduce or corroborate the contents of a web page. Web link mining systems usually look at both the quantity and type of links, often removing or decreasing the effect of local links since these tend to be navigational rather than referential.

The web can be seen as a graph (V, E) , where each vertex corresponds to a web page and each edge corresponds to a hyper-link. By using a predefined order among the vertices we can find a unique adjacency matrix corresponding to the web. Almost all web link mining algorithms use such an adjacency matrix, returning one or more eigenvectors corresponding to the eigenvalues of the adjacency matrix. Such eigenvectors can be seen as rating functions, giving a ranking or retrieval order for the corresponding pages.

Most of the research in web link mining has focused on variants of two algorithms called PageRank (see Section 3.2.1) and HITS (see Section 3.2.2).

3.2.1 PageRank

The general idea behind PageRank [26] is that of a *random surfer* browsing the web, at each time following a random link on the current web page. Given a sufficiently large number of simultaneous surfers, it would be possible to stop them at any given time and look at the number of surfers currently looking at each page and use that number as the relative probability that it is an important page.

There were some problems with this approach³, i.e. what to do when there are no outgoing links from a page and when two (or more) pages point to each other without outgoing links from the group (*rank sink*). The answer to the first problem was to recursively remove all pages lacking outgoing links from the calculations. The latter problem was countered by adding the possibility of jumping to any page on the web at a certain probability called a *damping factor* $(1 - \mu)$. The damping factor corresponds to the likelihood that a random surfer would jump to a random page rather than follow one of the links on current page. This value

²This can be seen in the proceedings from IJCAI Text-Mining & Link-Analysis workshop 2003 [62], LinkAnalysis-2005 [63], LinkKDD [3, 4, 5, 44], SIAM Workshop on Link Analysis, Counterterrorism and Security [13, 41, 124, 130], as well as papers published in other venues [56, 90, 110, 128, 131].

³Besides getting enough surfers to click at random.

must be between 0 (inclusive) and 1, and a value of 0.15 was used by the original authors [112]. The original PageRank algorithm gives a value for each page $j \in V$, which is obtained by solving Eq. (3.1) with $n = |V|$, using iteration to find a fixed point.

$$PR(j) = \frac{1-\mu}{n} + (\mu) \times \sum_{(i,j) \in E} PR(i)/\text{outdegree}(i) \quad (3.1)$$

This can also be described by using a matrix P obtained from the column-normalised adjacency matrix M (with all pages without links removed) of the graph (V, E) by adding the damping factor:

$$P = \left[\frac{1-\mu}{n} \right]_{n \times n} + \mu M \quad (3.2)$$

The rating returned, which is called PageRank, is the dominant eigenvector of P : $P\pi = \pi, \pi \geq 0, \|\pi\|_1 = 1$. This means that the i -th entry of π is the probability that a surfer visits page i , or the PageRank of page i .

Today n is between 15-20 billions and computing the eigenvector π was already in 2002 called the largest matrix computation problem in the world [104].

3.2.1.1 Rate of Convergence

It has been proved that the second largest eigenvalue of P will never be larger than μ [68], leading to fast convergence when using power iteration to find the PageRanks.⁴ It has also been shown that PageRank can achieve a stable state in $O(\log n)$ iterations, where n is the number of pages in the data-set. While this is sufficient for most applications, there have been a number of proposals for speeding up the calculations so it can be used for ranking large data-sets such as the entire Internet [10, 27, 39, 65, 66, 74, 82, 83, 84, 95, 114]. Typical examples of methods used for efficiency improvement include Arnoldi [121], Lanczos [60], Jacobi [23] and Gauss-Seidel [10].

⁴Because the power method converges at a rate proportional to $|\lambda_1/\lambda_2|$ [60] and P is an irreducible n -state Markov chain, which means that power iteration will always converge to a stable value [75, Theorem 5.2].

3.2.1.2 Problems and Variants

There are two main problems with the basic PageRank algorithm. The first problem is that there are huge computational costs involved in calculating the PageRank values once (described in the previous section). The second problem is that the values calculated represent an average random surfer rather than someone interested in one specific subject, thus potentially leading to an answer set that is not of interest for all users.

Two variations of PageRank have been widely used to counter the “randomness” problem. These are Personalized PageRank [112] and Topic-sensitive PageRank [67]. They both use the same general ideas and algorithm as the original PageRank, except that the damping factor is not added uniformly. Instead, a damping is scaled and added to either one starting page (for Personalized PageRank) or to a set of pages (for Topic-sensitive PageRank) assumed to be about that particular subject, which indicates that PageRank will have a preference for these pages over other pages. Personalized PageRank will thus give a view of the Internet from the viewpoint of one specific starting page.

Topic-sensitive PageRank has been used quite extensively, but suffers from a major problem when it comes to rate of convergence: Adding the damping factor to just some entries in the matrix makes it reducible. This means that several eigenvalues of the same magnitude might show up, thereby making the convergence of power iteration very slow [60]. This can partly be offset by using the approach of Jeh and Widom [77, 78], namely by creating base vectors for important pages. This corresponds to a partial view of the Internet according to each important page, by using Personalized PageRank with an extreme damping factor. The base vectors are scaled according to the corresponding eigenvalues and those that belong to the required set are aggregated and normalised in order to form the final answer vector. The rather small dampening factor used in PageRank still means that many iterations are required before a stable answer can be found for each base vector. Topic-sensitive PageRank is thus better for broader topics, so that each use of the vector can be seen as amortising the cost to generate it.

3.2.2 HITS

The basic idea behind HITS is that important pages about a specific subject have pages with links pointing to them, and pages with good links tend to point out important pages [88]. The algorithm gives two separate values for each page; how valuable the contained information is according to the algorithm (called *authority*) and also how good it is as a link page (called *hub*).

Rather than addressing the entire Internet directly it uses a bootstrap data-set, consisting of pages that are initially assumed to be about a specific subject. This set is further extended with all pages pointed to by the bootstrap set as well as pages that point to the bootstrap set. Each page in the entire set is given a start value in the two categories. These values are adjusted by simultaneous iteration over the equations given in Eq. (3.3), where η_i denotes the hub value for page i and α_j the authority value for page j .

$$\eta_i = \sum_{(i,j) \in E} \alpha_j \quad \alpha_j = \sum_{(i,j) \in E} \eta_i \quad (3.3)$$

Eq. (3.3) can also be described in terms of operations on the corresponding adjacency matrix A :

$$\eta = A^T \alpha = A^T A \eta \quad \alpha = A \eta = A A^T \alpha. \quad (3.4)$$

We remark that, in practice, the matrix products in Eq. (3.4) are never computed explicitly. All eigenvector-based methods only perform matrix-vector multiplications that make use of the sparse structure of the adjacency matrix A .

One thing to note here is that even though the required results are obtained in the relative differences between individual values in η and α , it is necessary to keep these values within $(0, 1)$ by using normalisation after each iteration of Eq. (3.4). These values can otherwise become so large as to cause overflows in calculations.

3.2.2.1 Rate of Convergence

The basic HITS algorithm usually has a very good convergence rate, since it could be seen as two simultaneous power iterations on symmetric non-negative matrices [60]. Using a bootstrap set also creates a data-set (and corresponding adjacency matrix A) that is *much* smaller than the entire Internet, leading to much faster evaluation of the hub and authority values.

3.2.2.2 Problems and Variants

HITS suffers from a problem called *topic drift*. Topic drift occurs when pages that are barely on-topic receive high hub and authority ratings, since these pages are a part of another close-knit society of pages linking to each other. This means that if more than one topic can be found within the extended data-set the one with the largest eigenvalue will be given. Possible solutions to this problem were given in the CLEVER project [32, 34] as well as the work of Bharat and Henzinger [21]. CLEVER uses different weights on the links depending on the the number of links and whether they reside on the same server, while Bharat and Henzinger used either outline filtering or dividing the weight of each link with the total number of links between same two servers.

The numerical stability of the calculations can sometimes be less than adequate, meaning that small changes (such as missing links) in the input data can change the focus from one cluster of pages to another. Possible solutions to this problem were given by Miller et al. [103] and Ng et al. [111]. Miller et al. used web logs and up to two link steps to generate the adjacency matrix, while Ng et al. used random walks in a manner similar to PageRank.

Another problem is that many different meanings of the same word can appear within the data-set. It is often the case that these meanings can be found by checking more than the first eigenvalues for the combination, using what is called spectral graph theory [92, 111].

3.3 Thesis Contributions

We have used two major approaches to obtain a web search system that is stable, fast and, according to the users, returns good answer vectors.

3.3.1 Monotone Data Flow System

The first approach is to put *trust levels*⁵ on the meta-data belonging to a page and then propagating it along links. The propagation is controlled by

- hyper-links (either given explicitly in the web pages or implied by the paths of the URLs),
- the trust level given to each page,
- whether the data was perceived as pervasive, i.e. should propagate more than one link, and
- a function that calculates the resulting meta-data using the incoming values from each link.

This corresponds to a weighted Topic-sensitive PageRank where each non-pervasive value can be propagated just one step along the links, and all links have weight. The approach builds on the work done by Kam and Ullman [80], with an updated propagation step to fit the requirements of our model.

The prototype is quite slow and requires inside knowledge to be used successfully; well-defined trust rules as well as a relatively small input data-set are essential. Experiments with the prototype gave very positive results, even though both the model and the resulting search engine are more of a theoretical and academic, rather than a practical, nature [142, Paper IV].

⁵How much trust we put in that page regarding each piece of meta-data.

3.3.2 Propagation of Topic-Relevance

The Propagation of Topic-Relevance⁶ (ProT) algorithm is a close relative to Topic-sensitive PageRank, but with a major change. All links used in the calculations potentially have the same strength; the value to propagate is divided by the *decay factor* (ξ) rather than dividing the value to propagate among the outgoing links as in PageRank. This means that the propagation step requires a little less work, but it does require both a very carefully chosen ξ and normalisation after each iteration [144, Paper VI].

Given an initial score $\varpi(j, 0) = 1$ (100%) for pages that are assumed to be on-topic and zero otherwise, and using k as the iteration count as well as setting ξ to an appropriate value (i.e. just above the dominant eigenvalue of the adjacency matrix) we can apply the following algorithm:

$$\varpi(j, k) = \frac{1}{\xi} \sum_{(i,j) \in E} \varpi(i, k-1) + \begin{cases} \varpi(j, k-1) & \text{if } j \text{ is on-topic} \\ 0 & \text{otherwise.} \end{cases} \quad (3.5)$$

The final answer is given after normalisation of the k :th ϖ vector.

This means that the final answer depends on both the links of the web and which pages are on-topic, controlled by the choice of ξ . This corresponds to changing the value of the damping factor of Topic-sensitive PageRank, albeit using a value for the damping factor that is far away from the usual choices.

⁶The name was originally Propagation of Trust [143, Paper V].

3.3.2.1 Problems and Variants

The matrix that ProT operates on is a composition of the adjacency matrix of the original web and self-referential links for all pages that are on-topic. The problem with this matrix is that it is reducible, meaning that the matrix might have several eigenvalues of the same magnitude. This leads to *very* slow convergence when using the power method to find the dominant eigenvalue (and corresponding eigenvector) of the matrix. The convergence rate of ProT is on the same order of magnitude as for the original Topic-sensitive PageRank, as we have shown [144, Paper VI, Section 11.5.1].

Our solution to this problem is to look at one starting page at a time, then adding up all resulting vectors (called *basic vectors*) to generate a final result vector (after normalisation). This also has the advantage that larger values of ξ can be chosen, leading to even faster convergence. We call this version *Superpositioned Singleton Propagation of Topic-Relevance* (S²ProT) [144, Paper VI, Section 11.4.4].

Another solution is the *Hybrid Superpositioned Singleton Propagation of Topic-Relevance* (HyS²ProT) algorithm, using the same general idea as S²ProT but diminishing each propagated value further by dividing the value with the number of outgoing links, in the same manner as in PageRank [144, Paper VI, Section 11.7.2]. The main advantage of this approach is that the matrix has a dominant eigenvalue of 1, since it uses a normalised matrix in the same way as PageRank (see Section 3.2.1 on pages 17–18). This also means that an even larger value of ξ must be chosen, since the starting values will otherwise propagate further along the links.

3.3.3 Evaluation of Empirical Result

We have used three different ways of evaluating the algorithms:

- Empirical assessment of result relevance using human graders. In Paper V [143] we took the top pages given by each algorithm and added them to a questionnaire for each chosen search term. Volunteer graders graded each page according to its perceived relevance with respect to the search term. The average of all valid answers⁷ of pages belonging to the top pages of each algorithm was calculated, and compared with the results from the others.
This assessment method was reused with minor changes in Paper VI⁸. All results indicate that our algorithms (especially S²ProT) yield good answer sets according to the graders [144, Paper VI, Section 11.6.1.4].
- Experimental assessment of algorithm stability.
 - The stability of each algorithm when removing pages from the set of on-topic pages were tested. The results indicated that S²ProT were more stable than Topic-sensitive PageRank, which in turn was more stable than ProT [144, Paper VI, Section 11.6.2.1].
 - The stability of each algorithm when removing links from the dataset was tested. The results show that our algorithms are very stable; the ranking order between the algorithms varies slightly depending on which measurement we use [144, Paper VI, Section 11.6.2.2].

⁷Ignoring grades of “Don’t know” [143, Paper V, Section 10.2.2 on page 126].

⁸See Section 11.6.1.3 on page 168.

3.4 Summary

We have made extensive qualitative studies in various aspects of the algorithms described in this chapter, presented in Table 3.1 on the next page. Some of the results were discussed in Section 3.3.3 on the preceding page and have already been published [144, Paper VI, Sections 11.5 to 11.6], while others (specifically some of the HITS data) are based on data found in other sources [88, 103, 111].

We have graded each algorithm on a relative scale from ‘+’ to ‘++++’ with regards to scalability, stability, and relevance. More plus signs correspond to a higher grade. We remark that this grading reflects a qualitative assessment of the figures revealed by our tests, but that the plus signs are not directly comparable between columns. This means that one should not compare the algorithms by adding up all the plus signs directly.

For *scalability*, we have compared the cost of using larger input data-sets [144, Paper VI, Section 11.5]. It reflects both the rate of convergence and the memory requirements. The most scalable algorithms are PageRank⁹ and S²ProT, followed by HyS²ProT, and then the others. One could argue that HITS should have a slightly higher grade because of its diminished data-set, but actual data does not agree with that; The data-set must not only be generated from the larger set but the generated set will sometimes have several eigenvalues of the same magnitude as well, which indicates that we could not give it a higher grade. Using the algorithm upgrade of Jeh and Widom [77, 78] would take Topic-sensitive PageRank up to the same level as HyS²ProT.

Stability indicates how much the results are affected by removal of links and decreased sets of starting pages [144, Paper VI, Section 11.6.2]. We have also performed the same tests using HITS, and the results agree with the data reported by Ng et al. [111], i.e. HITS is quite unstable.

Assessment of perceived *relevance* has been one of the major parts of our work in both Paper V and Paper VI. We have chosen to group algorithms with similar results (see [143, Paper V] and [144, Paper VI, Section 11.6.1]) to the same grade, although there are minor differences within the groups. The higher the perceived relevance, the more plus signs are given.

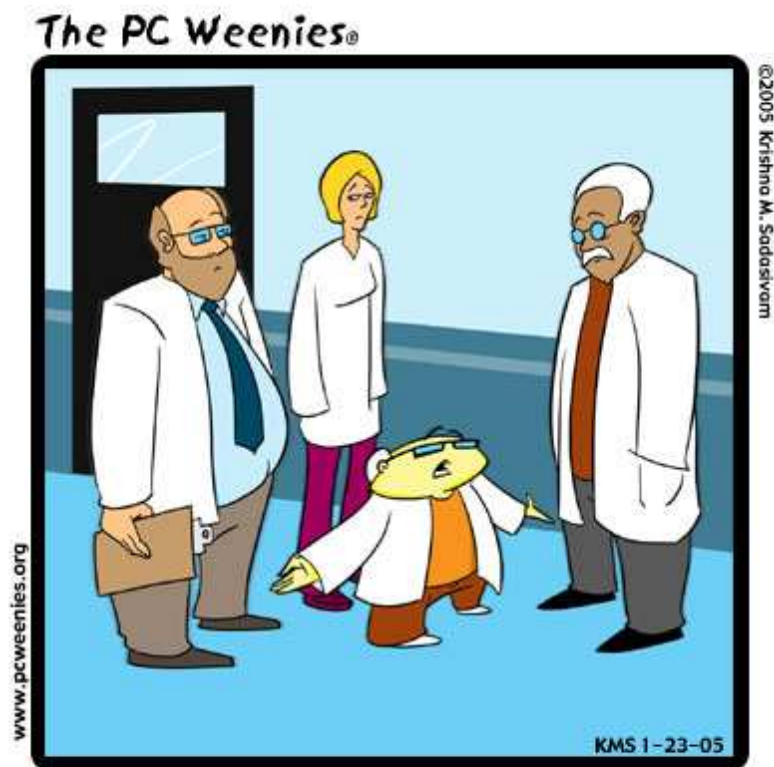
Our conclusion is that the S²ProT algorithm is among the best in all categories.

⁹But recall, unlike the other algorithms in the table, PageRank is not topic-sensitive.

Table 3.1: The scalability, stability and relevance of each algorithm on a scale from ‘+’ to ‘++++’.

Algorithm	Scalability	Stability	Relevance
<i>PageRank</i>	++++	++++	+++
<i>Personalized PageRank</i>	++	++++	+
<i>Topic-sensitive PageRank</i>	++	+++	++++
<i>HITS^a</i>	++	+	+++
<i>ProT</i>	++	++	+++
<i>S²ProT</i>	++++	++++	++++
<i>HyS²ProT</i>	+++	++++	++++

^aBased partly on the results from other sources.



“UNFORTUNATELY, THE RESULTS OF OUR
STUDY ON AMBIGUOUS SEARCH TERMS
PROVED TO BE INCONCLUSIVE.”

Figure 3.1: The problem with ambiguous search terms.

With permission from Krishna M. Sadasivam.

Chapter 4

Final Remarks

The work described in this thesis can be seen as a set of algorithms and their implementations that all operate on large quantities of discrete and textual data. The general idea is that the information is sampled, extracted, compiled and stored in a central data base that can be accessed by all tools that require the information.

Figure 4.1 on the next page illustrates our view of how such a set of tools should be interconnected. Documents to be included in the data base are processed to extract relevant information and possibly meta-data. Data propagation or implication can be performed if some documents lack sufficient data, e.g. [142, Paper IV].

Another possible source of data is a label bureau that provides clients with meta-data information about documents [14, 91, 102, 118, 138].

Multiple back-ends exist for the system as we envision it, one being a data mining system that mines for association rules. It uses CHiC [140, 141, Papers II–III] as the first step to create concept hierarchies, used in later association rule mining.

Another available back-end tool is a search engine that uses the topic-specific vectors created by our search engine algorithms [144, Paper VI] in order to facilitate searching. A prototype of a complete web-based search engine has been created and tested.

All in all, the algorithms and tools described in this thesis work together to provide answers that each of them would not be able to answer on their own. Most of the individual tools in Figure 4.1 on the following page already exist, but they have not been integrated into a framework or system.

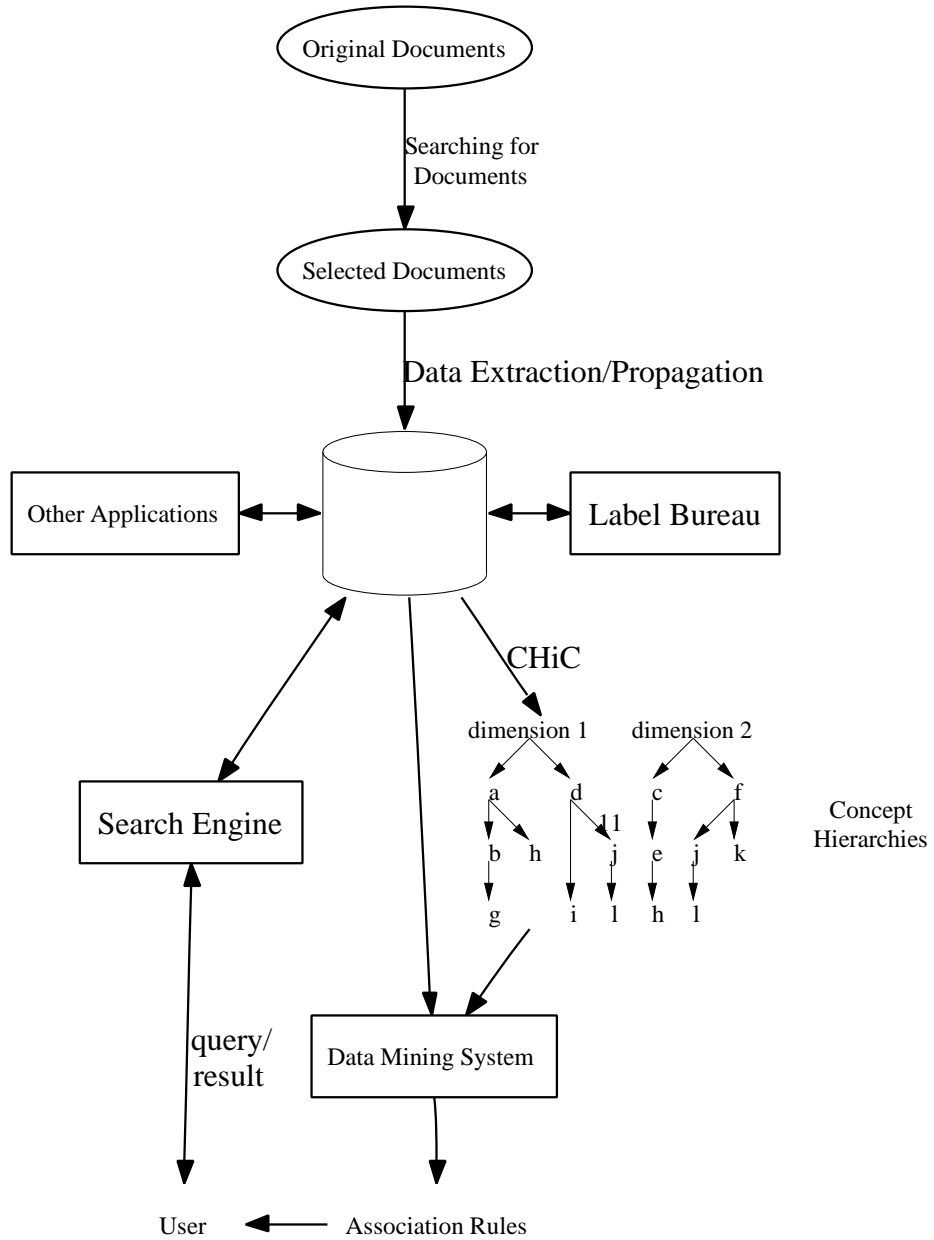


Figure 4.1: Overview of the application environment of our view of a data mining and management system for discrete and textual data.

Chapter 5

Bibliography

- [1] ABDULJALEEL, N., AND QU, Y. Domain term extraction and structuring via link analysis. In Grobelnik et al. [63].
- [2] ACHARYYA, S., AND GHOSH, J. A maximum entropy framework for higher order link analysis on directed graphs. In Donoho et al. [44].
- [3] ADIBI, J., CHALUPSKY, H., GROBELNIK, M., MILIC-FRAYLING, N., AND MLADENIC, D., Eds. *Workshop on Link Analysis and Group Detection (LinkKDD2004)* (Seattle, WA, USA, Aug. 22, 2004). See [6, 22, 33, 37, 57, 79, 97, 106, 115, 116, 117].
- [4] ADIBI, J., GROBELNIK, M., MILIC-FRAYLING, N., MLADENIC, D., AND PANTEL, P., Eds. *Workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD2006)* (Philadelphia, PA, USA, Aug. 20, 2006). See [11, 16, 40, 70, 87, 123].
- [5] ADIBI, J., GROBELNIK, M., MLADENIC, D., AND PANTEL, P., Eds. *Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD2005)* (Chicago, IL, USA, Aug. 21, 2005). See [12, 29, 50, 69, 98, 122, 127, 135].
- [6] ADIBI, J., MORRISON, C. M., AND COHEN, P. R. Measuring confidence intervals in link discovery: A bootstrap approach. In Adibi et al. [3].
- [7] AL HASAN, M., CHAOJI, V., SALEM, S., AND ZAKI, M. Link prediction using supervised learning. In Teredesai and Carley [130].
- [8] AMERICAN LIBRARY OF CONGRESS. Electronic CIP: Cataloging in publication program. Web site, Oct. 06, 2008. Date visited given, <http://cip.loc.gov/>.
- [9] AMSTERDAM INTERNET EXCHANGE. AMS-IX. Web site, Sept. 26, 2008. Date visited given, <http://www.ams-ix.net/>.
- [10] ARASU, A., NOVAK, J., TOMKINS, A., AND TOMLIN, J. PageRank computation and the structure of the web: Experiments and algorithms. Tech. rep., IBM Almaden Research Center, Nov. 2001.
- [11] ASUR, S., PARTHASARATHY, S., AND UCAR, D. An ensemble approach for clustering scale-free graphs. In Adibi et al. [4].
- [12] BADIA, A., AND KANTARDZIC, M. Graph building as a mining activity: Finding links in the small. In Adibi et al. [5].

- [13] BADIA, A., AND SKILLICORN, D., Eds. *Workshop on Link Analysis, Counterterrorism and Security (Adversarial Data Analysis)* (2008). See [17, 105].
- [14] BAIRD-SMITH, A. *Jigsaw: An object oriented server*. The World Wide Web Consortium, Cambridge, Massachusetts, Feb. 1997.
- [15] BATAGELJ, V., AND MRVAR, A. Density based approaches to network analysis: Analysis of reuters terror news network. In Donoho et al. [44].
- [16] BECCHETTI, L., CASTILLO, C., DONATO, D., AND FAZZONE, A. Comparison of sampling techniques for web graph characterization. In Adibi et al. [4].
- [17] BEER, E. A., PRIEBE, C. E., AND SCHEINERMAN, E. R. Torus graph inference for detection of localized activity. In Badia and Skillicorn [13].
- [18] BEN-DOV, M., WU, W., FELDMAN, R., AND CAIRNS, P. A. Improving knowledge discovery by combining text-mining & link analysis techniques. In Cybenko and Srivastava [41].
- [19] BERRY, M. W., AND BROWNE, M. Email surveillance using nonnegative matrix factorization. In Skillicorn and Carley [124], pp. 45–54.
- [20] BERTINO, E., CATANIA, B., AND ZARRI, G. P. *Intelligent Database Systems*. Pearson Education, 2001.
- [21] BHARAT, K., AND HENZINGER, M. R. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1998), ACM Press, pp. 104–111.
- [22] BHATTACHARYA, I., AND GETOOR, L. Deduplication and group detection using links. In Adibi et al. [3].
- [23] BIANCHINI, M., GORI, M., AND SCARSELLI, F. Inside PageRank. *ACM Trans. Inter. Tech.* 5, 1 (2005), 92–128.
- [24] BLOEDORN, E., ROTHLEDER, N. J., DEBARR, D., AND ROSEN, L. Relational graph analysis with real-world constraints: An application in irs tax fraud detection. In Grobelnik et al. [63].
- [25] BORGES, J., AND LEVENE, M. Ranking pages by topology and popularity within web sites. *World Wide Web* 9, 3 (2006), 301–316.
- [26] BRIN, S., AND PAGE, L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* 30, 1–7 (1998), 107–117.
- [27] BRODER, A. Z., LEMPEL, R., MAGHOUL, F., AND PEDERSEN, J. Efficient PageRank approximation via graph aggregation. *Information Retrieval* 9, 2 (Mar. 2006), 123–138.
- [28] BUNTINE, W., LÖFSTRÖM, J., PERTTU, S., AND VALTONEN, K. Topic-specific link analysis using independent components for information retrieval. In Grobelnik et al. [63].
- [29] CAI, D., SHAO, Z., HE, X., YAN, X., AND HAN, J. Mining hidden community in heterogeneous social networks. In Adibi et al. [5].
- [30] CARACCILO, C., DE RIJKE, M., AND KIRCZ, J. Towards scientific information disclosure through concept hierarchies. In *Proceedings ELPUB 2002* (2002).
- [31] CHAKRABARTI, D., ZHAN, Y., BLANDFORD, D., FALOUTSOS, C., AND BLELLOCH, G. NetMine: mining tools for large graphs. In Cybenko and Srivastava [41].
- [32] CHAKRABARTI, S. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *WWW '01: Proceedings of the 10th international conference on World Wide Web* (New York, NY, USA, 2001), ACM Press, pp. 211–220.

- [33] CHAKRABARTI, S. Discovering links between lexical and surface features in questions and answers. In Adibi et al. [3].
- [34] CHAKRABARTI, S., DOM, B. E., AND INDYK, P. Enhanced hypertext categorization using hyperlinks. In *Proceedings of SIGMOD-98, ACM International Conference on Management of Data* (Seattle, US, 1998), L. M. Haas and A. Tiwary, Eds., ACM Press, New York, US, pp. 307–318.
- [35] CHAPANOND, A., KRISHNAMOORTHY, M. S., AND YENER, B. Graph theoretic and spectral analysis of enron email data. In Skillicorn and Carley [124], pp. 15–22.
- [36] CHIDLOVSKII, B., RAGETLI, J., AND DE RIJKE, M. Wrapper generation via grammar induction. In *Proceedings European Conference on Machine Learning (ECML'2000)* (2000), LNCS, Springer.
- [37] CHKLOVSKI, T., AND PANTEL, P. Path analysis for refining verb relations. In Adibi et al. [3].
- [38] CLÉROT, F., AND NGUYEN, Q. A social network approach for the ranking of the autonomous systems of the internet. In Grobelnik et al. [63].
- [39] CORSO, G. M. D., GULLI, A., AND ROMANI, F. Fast PageRank computation via a sparse linear system. In *Proceedings of Third Workshop on Algorithms and Models for the Web-Graph (WAW 2004)* (Rome, Italy, Oct. 16, 2004).
- [40] CREAMER, G., AND STOLFO, S. A link mining algorithm for earnings forecast using boosting. In Adibi et al. [4].
- [41] CYBENKO, G. V., AND SRIVASTAVA, J., Eds. *Workshop on Link Analysis, Counterterrorism and Security* (2004). See [18, 31, 49].
- [42] DAVISON, B. D. Topical locality in the web. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (Athens, Greece, 2000), ACM Press, pp. 272–279.
- [43] DAVISON, B. D. Unifying text and link analysis. In Grobelnik et al. [62].
- [44] DONOHO, S., DYBALA, T., GROBELNIK, M., MILIC-FRAYLING, N., AND MLADENIC, D., Eds. *Proceedings of the 2003 Link Analysis for Detecting Complex Behavior (LinkKDD2003) Workshop* (Washington, DC, USA, Aug. 27, 2003). See [2, 15, 58, 85, 132].
- [45] DUAN, Y., WANG, J., KAM, M., AND CANNY, J. A secure online algorithm for link analysis on weighted graph. In Skillicorn and Carley [124], pp. 71–81.
- [46] DUBLIN CORE METADATA INITIATIVE. *The Dublin Core Metadata Element Set*. ISO Standard 15836 (2003) and ANSI/NISO Standard Z39.85-2007.
- [47] DUNHAM, M. H. *Data Mining, Introductory and Advanced Topics*. Prentice Hall, inc., Englewood Cliffs, New Jersey, 2003.
- [48] FAGNI, T., PEREGO, R., SILVESTRI, F., AND ORLANDO, S. Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data. *ACM Trans. Inf. Syst.* 24, 1 (2006), 51–78.
- [49] FALOUTSOS, C., MCCURLEY, K. S., AND TOMKINS, A. Connection subgraphs in social networks. In Cybenko and Srivastava [41].
- [50] FISSAHA ADAFRE, S., AND DE RIJKE, M. Discovering missing links in wikipedia. In Adibi et al. [5].

- [51] FISSAHA ADAFRE, S., JIJKOUN, V., AND DE RIJKE, M. Link-based vs. content-based retrieval for question answering using wikipedia. In *Evaluation of Multilingual and Multimodal Information Retrieval (2007)*, pp. 537–540.
- [52] FOX, J. Webless Literate Programming. *TUGboat 11*, 4 (Nov. 1990).
- [53] FREITAG, D. Information extraction from HTML: Application of a general machine learning approach. In *AAAI/IAAI (1998)*, pp. 517–523.
- [54] FRIENDLY, L. The Design of Distributed Hyperlinked Programming Documentation. In *Proceedings of the 1995 International Workshop on Hypermedia Design (June 1995)*.
- [55] GANIZ, M. C., POTTENGER, W. M., AND YANG, X. Link analysis of higher-order paths in supervised learning datasets. In Teredesai and Carley [130].
- [56] GETOOR, L. Link mining: a new data mining challenge. *SIGKDD Explor. Newsl.* 5, 1 (2003), 84–89.
- [57] GILBERT, A. C., AND LEVCHENKO, K. Compressing network graphs. In Adibi et al. [3].
- [58] GOLDENBERG, A., KUBICA, J., AND KOMAREK, P. A comparison of statistical and machine learning algorithms on the task of link completion. In Donoho et al. [44].
- [59] GOLDENBERG, A., AND MOORE, A. Empirical bayes screening for link analysis. In Grobelnik et al. [62].
- [60] GOLUB, G. H., AND VAN LOAN, C. F. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996.
- [61] GREGG, D. G., AND WALCZAK, S. Adaptive web information extraction. *Commun. ACM* 49, 5 (2006), 78–84.
- [62] GROBELNIK, M., MILIC-FRAYLING, N., AND MLADENIC, D., Eds. *Proceedings of the 2003 IJCAI Text-Mining & Link-Analysis Workshop (Acapulco, Mexico, Aug. 9, 2003)*. See [43, 59, 93, 99].
- [63] GROBELNIK, M., MILIC-FRAYLING, N., AND MLADENIC, D., Eds. *The AAAI-05 Workshop on Link Analysis (LinkAnalysis-2005) (July 10 2005)*. See [1, 24, 28, 38, 107, 113, 136].
- [64] HAN, J., AND KAMBER, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 2001.
- [65] HAVELIWALA, T., KAMVAR, S., KLEIN, D., MANNING, C., AND GOLUB, G. Computing PageRank using power extrapolation. Tech. rep., Stanford University, CA, USA, Oct. 18, 2003.
- [66] HAVELIWALA, T. H. Efficient computation of PageRank. Tech. Rep. 1999-31, Stanford University Database Group, Oct. 18, 1999.
- [67] HAVELIWALA, T. H. Topic-sensitive PageRank. In *Proceedings of the eleventh international conference on World Wide Web (2002)*, ACM Press, pp. 517–526.
- [68] HAVELIWALA, T. H., AND KAMVAR, S. D. The second eigenvalue of the google matrix. Tech. rep., Stanford University, Mar. 2003.
- [69] HILL, S., AGARWAL, D., BELL, R., AND VOLINSKY, C. Tuning representations of dynamic network data. In Adibi et al. [5].
- [70] HUANG, Z. Link prediction based on graph topology: The predictive value of generalized clustering coefficient. In Adibi et al. [4].
- [71] HUMPHRIES, M., HAWKINS, M. W., AND DY, M. C. *Data Warehousing: Architecture and Implementation*. Prentice Hall PTC, Upper Saddle River, New Jersey, 1999.

-
- [72] IEEE LEARNING TECHNOLOGY STANDARDS COMMITTEE. *IEEE 1484.12.1-2002 Standard for Learning Object Metadata*, 2002.
- [73] INTERNET SYSTEMS CONSORTIUM, INC. ISC Domain Survey: Number of Internet Hosts. Web page, Sept. 25, 2008. Date visited given, <http://www.isc.org/index.pl?ops/ds/host-count-history.php>.
- [74] IPSEN, I. C. F., AND KIRKLAND, S. Convergence analysis of a PageRank updating algorithm by Langville and Meyer. *SIAM J. Matrix Anal. Appl.* 27, 4 (2006), 952–967.
- [75] IPSEN, I. C. F., AND MEYER, C. D. Uniform stability of markov chains. *SIAM J. Matrix Anal. Appl.* 15, 4 (1994), 1061–1074.
- [76] JAIN, A. K., MURTY, M. N., AND FLYNN, P. J. Data Clustering: A Review. *ACM Computing Surveys (CSUR)* 31, 3 (Sept. 1999), 264–323.
- [77] JEH, G., AND WIDOM, J. Scaling personalized web search. Tech. Rep. 2002-12, Stanford University Database Group, 2002.
- [78] JEH, G., AND WIDOM, J. Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web* (New York, NY, USA, 2003), ACM Press, pp. 271–279.
- [79] JONES, R. Semisupervised learning on small worlds. In Adibi et al. [3].
- [80] KAM, J. B., AND ULLMAN, J. D. Global data flow analysis and iterative algorithms. *Journal of the ACM (JACM)* 23, 1 (1976), 158–171.
- [81] KAMPS, J., MONZ, C., DE RIJKE, M., AND SIGURBJÖRNSSON, B. Approaches to robust and web retrieval. In *Proceedings TREC 2003* (2004), pp. 594–600.
- [82] KAMVAR, S. D., HAVELIWALA, T. H., AND GOLUB, G. H. Adaptive methods for the computation of PageRank. Tech. rep., Stanford University, CA, USA, Apr. 2003.
- [83] KAMVAR, S. D., HAVELIWALA, T. H., MANNING, C. D., AND GOLUB, G. H. Exploiting the block structure of the web for computing PageRank. Tech. rep., Stanford University, CA, USA, Mar. 4, 2003.
- [84] KAMVAR, S. D., HAVELIWALA, T. H., MANNING, C. D., AND GOLUB, G. H. Extrapolation methods for accelerating PageRank computations. In *Proceedings of the Twelfth International World Wide Web Conference* (2003).
- [85] KARGUPTA, H., LIU, K., DATTA, S., RYAN, J., AND SIVAKUMAR, K. Link analysis, privacy preservation, and random perturbations. In Donoho et al. [44].
- [86] KEILA, P. S., AND SKILLICORN, D. B. Structure in the enron email dataset. In Skillicorn and Carley [124], pp. 55–64.
- [87] KETKAR, N. S., HOLDER, L. B., AND COOK, D. J. Mining in the proximity of subgraphs. In Adibi et al. [4].
- [88] KLEINBERG, J. Authoritative sources in a hyperlinked environment. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms* (1998), pp. 668–677.
- [89] KOLDA, T., AND BADER, B. The TOPHITS model for higher-order web link analysis. In Teredesai and Carley [130].
- [90] KOSALA, AND BLOCCKEEL. Web mining research: A survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM 2* (2000).

- [91] KRAUSKOPF, T., MILLER, J., RESNICK, P., AND TREESE, W. *REC-PICS-labels-961031: PICS Label Distribution Label Syntax and Communication Protocols*. The World Wide Web Consortium, Cambridge, Massachusetts, Oct. 31, 1996.
- [92] KROEKER, K. L. Finding diamonds in the rough. *Communications of the ACM* 51, 9 (Sept. 2008), 11–13.
- [93] KUBICA, J., MOORE, A., COHN, D., AND SCHNEIDER, J. cGraph: A fast graph-based method for link analysis and queries. In Grobelnik et al. [62].
- [94] LANDAUER, T. K., FOLTZ, P. W., AND LAHAM, D. Introduction to latent semantic analysis. *Discourse Processes* 25 (1998), 259–284.
- [95] LANGVILLE, A. N., AND MEYER, C. D. Updating PageRank using the group inverse and stochastic complementation. Tech. Rep. CRSC-TR02-32, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, USA, Nov. 2002.
- [96] LEHMANN, S. Live and dead nodes. In Skillicorn and Carley [124], pp. 65–70.
- [97] LESKOVEC, J., GROBELNIK, M., AND MILIC-FRAYLING, N. Learning sub-structures of document semantic graphs for document summarization. In Adibi et al. [3].
- [98] LICAMELE, L., BILGIC, M., GETOOR, L., AND ROUSSOPOULOS, N. Capital and benefit in social networks. In Adibi et al. [5].
- [99] LU, Q., AND GETOOR, L. Link-based text classification. In Grobelnik et al. [62].
- [100] MADHAVAN, J., KO, D., KOT, Ł., GANAPATHY, V., RASMUSSEN, A., AND HALEVY, A. Google’s deep web crawl. *Proc. VLDB Endow.* 1, 2 (2008), 1241–1252.
- [101] MARSHALL, C. C. Making metadata: a study of metadata creation for a mixed physical-digital collection. In *DL ’98: Proceedings of the third ACM conference on Digital libraries* (New York, NY, USA, 1998), ACM Press, pp. 162–171.
- [102] MILLER, J., RESNICK, P., AND SINGER, D. *REC-PICS-services-961031: Rating Services and Rating Systems (and Their Machine Readable Descriptions)*. The World Wide Web Consortium, Cambridge, Massachusetts, Oct. 31, 1996.
- [103] MILLER, J. C., RAE, G., SCHAEFER, F., WARD, L. A., LOFARO, T., AND FARAHAT, A. Modifications of Kleinberg’s HITS algorithm using matrix exponentiation and web log records. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (New Orleans, Louisiana, United States, 2001), ACM Press, pp. 444–445.
- [104] MOLER, C. B. Cleve’s corner: The world’s largest matrix computation: Google’s pagerank is an eigenvector of a matrix of order 2.7 billion. Technical note, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, Oct. 2002.
- [105] MOON, I.-C., CARLEY, K. M., AND LEVIS, A. H. Vulnerability assessment on adversarial organization: Unifying command and control structure analysis and social network analysis. In Badia and Skillicorn [13].
- [106] MUKHERJEE, M., AND HOLDER, L. B. Graph-based data mining on social networks. In Adibi et al. [3].
- [107] MURRAY, K., HARRISON, I., LOWRANCE, J., RODRIGUEZ, A., THOMERE, J., AND WOLVERTON, M. Pherl: An emerging representation language for patterns and hypotheses and evidence. In Grobelnik et al. [63].

- [108] MUSLEA, I., MINTON, S., AND KNOBLOCK, C. A. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems* 4, 1/2 (2001), 93–114.
- [109] NESTOROV, S., ABITEBOUL, S., AND MOTWANI, R. Extracting schema from semistructured data. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 1998), ACM Press, pp. 295–306.
- [110] NG, A. Y., ZHENG, A. X., AND JORDAN, M. Link analysis, eigenvectors, and stability. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)* (2001).
- [111] NG, A. Y., ZHENG, A. X., AND JORDAN, M. Stable algorithms for link analysis. In *Proceedings of the Twenty-fourth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Sept. 2001).
- [112] PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. The PageRank citation ranking: Bringing order to the web. Tech. rep., Stanford Digital Library Technologies Project, 1998.
- [113] PAPERINICK, N., AND HAUPTMANN, A. G. Summarization of broadcast news video through link analysis of named entities. In Grobelnik et al. [63].
- [114] PARREIRA, J. X., CASTILLO, C., DONATO, D., MICHEL, S., AND WEIKUM, G. The Juxtaposed approximate PageRank method for robust PageRank approximation in a peer-to-peer web search network. *The International Journal on Very Large Data Bases* 17, 2 (Mar. 2008), 291–313.
- [115] PIOCH, N. J., HUNTER, D., WHITE, J. V., KAO, A., BOSTWICK, D., AND JONES, E. K. Multi-hypothesis abductive reasoning for link discovery. In Adibi et al. [3].
- [116] RAGHAVAN, H., ALLAN, J., AND MCCALLUM, A. An exploration of entity models, collective classification and relation description. In Adibi et al. [3].
- [117] RESIG, J., DAWARA, S., HOMAN, C. M., AND TEREDESAI, A. Extracting social networks from instant messaging populations. In Adibi et al. [3].
- [118] RESNICK, P., AND MILLER, J. PICS: Internet Access Controls Without Censorship. *Communications of the ACM* 39, 10 (1996), 87–93.
- [119] SALTON, G., WONG, A., AND YANG, C. S. A vector space model for automatic indexing. *Communications of the ACM* 18, 11 (Nov. 1975), 613–620.
- [120] SANDERSON, M., AND CROFT, B. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (Berkeley, California, United States, 1999), ACM Press, pp. 206–213.
- [121] SCOTT, J. A. An Arnoldi code for computing selected eigenvalues of sparse, real, unsymmetric matrices. *ACM Trans. Math. Softw.* 21, 4 (1995), 432–475.
- [122] SHETTY, J., AND ADIBI, J. Discovering important nodes through graph entropy - the case of enron email database. In Adibi et al. [5].
- [123] SIDIROPOULOS, A., KATSAROS, D., AND MANOLOPOULOS, Y. Generalized h-index for revealing latent facts in social networks of citations. In Adibi et al. [4].
- [124] SKILLICORN, D., AND CARLEY, K., Eds. *Workshop on Link Analysis, Counterterrorism and Security* (2005). See [19, 35, 45, 86, 96].
- [125] SODERLAND, S. Learning information extraction rules for semi-structured and free text. *Machine Learning* 34, 1-3 (1999), 233–272.

- [126] SODERLAND, S., FISHER, D., ASELTINE, J., AND LEHNERT, W. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (San Francisco, 1995), C. Mellish, Ed., Morgan Kaufmann, pp. 1314–1319.
- [127] STOILOVA, L., HOLLOWAY, T., MARKINES, B., MAGUITMAN, A., AND MENCZER, F. Givealink: Mining a semantic network of bookmarks for web search and recommendation. In Adibi et al. [5].
- [128] TAN, P.-N., AND KUMAR, V. Mining indirect associations in web data. In *Proceedings of the 2002 Mining Log Data Across All Customer TouchPoints (WebKDD2001) Workshop* (Aug. 2001).
- [129] TAN, P.-N., STEINBACH, M., AND KUMAR, V. *Introduction to Data Mining*. Addison-Wesley, Reading, Massachusetts, 2006.
- [130] TEREDesai, A., AND CARLEY, K., Eds. *Workshop on Link Analysis, Counterterrorism and Security* (2006). See [7, 55, 89].
- [131] TIAN, Y., HUANG, T., AND GAO, W. A web site mining algorithm using the multiscale tree representation model. In *Proceedings of the 2003 Webmining as a Premise to Effective and Intelligent Web Applications (WebKDD'2003) Workshop* (Aug. 2003), pp. 83–92.
- [132] TIAN, Y., MEI, Z., HUANG, T., AND GAO, W. Incremental learning for interaction dynamics with the influence model. In Donoho et al. [44].
- [133] TWO CROWS CORPORATION. *Introduction to Data Mining and Knowledge Discovery*, Third Edition. Potomac, MD, USA, 2005.
- [134] VAN HEESCH, D. *doxygen: Manual for version 1.5.7.1*, 2008. Available for download from ftp://ftp.stack.nl/pub/users/dimitri/doxygen_manual-1.5.7.1.pdf.zip.
- [135] WANG, X., MOHANTY, N., AND MCCALLUM, A. Group and topic discovery from relations and text. In Adibi et al. [5].
- [136] WOLVERTON, M., AND THOMERE, J. The role of higher-order constructs in the inexact matching of semantic graphs. In Grobelnik et al. [63].
- [137] XIAO, Y., AND DUNHAM, M. H. Efficient mining of traversal pattern. *Data and Knowledge Engineering* 39, 2 (Nov. 2001), 191–214.
- [138] ÅGREN, O. Reuse via the World Wide Web: How to Find the Software Required for Reuse. Master's thesis, Umeå University, Umeå, Sweden, Dec. 1998. UMNAD 242.98.
- [139] ÅGREN, O. ALGEXT - an ALGORITHM EXTRACTOR for C Programs. Tech. rep., Umeå University, Umeå, Sweden, May 2001. UMINF 01.11, ISSN 0348-0542, Paper I on page 41.
- [140] ÅGREN, O. Automatic Generation of Concept Hierarchies for a Discrete Data Mining System. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE '02)* (Las Vegas, Nevada, USA, June 24-27, 2002), pp. 287–293. Paper II on page 59.
- [141] ÅGREN, O. CHIC: A Fast Concept Hierarchy Constructor for Discrete or Mixed Mode Databases. In *Proceedings of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)* (San Francisco, California, USA, July 1-3, 2003), pp. 250–258. Paper III on page 77.
- [142] ÅGREN, O. Propagation of Meta Data over the World Wide Web. In *Proceedings of the International Conference on Internet Computing (IC '03)* (Las Vegas, Nevada, USA, June 23-26, 2003), vol. 2, pp. 670–676. Paper IV on page 103.

-
- [143] ÅGREN, O. Assessment of WWW-Based Ranking Systems for Smaller Web Sites. *INFO-COMP Journal of Computer Science* 5, 2 (June 2006), 45–55. Paper V on page 117.
- [144] ÅGREN, O. S²ProT: Rank Allocation by Superpositioned Propagation of Topic-Relevance. *International Journal of Web Information Systems* 4, 4 (2008), 416–440. Paper VI on page 141.

