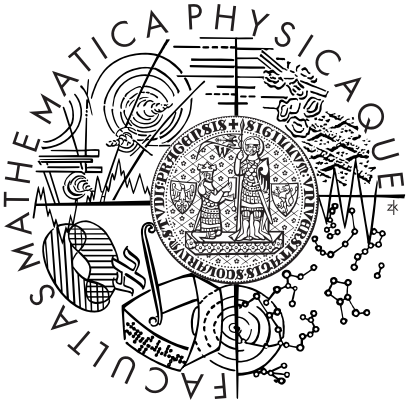
Charles University Prague

Faculty of Mathematics and Physics

DOCTORAL THESIS

Václav Petříček

Quantitative analysis of networked environments to improve performance of information systems



Department of Software Engineering

Advisor: Prof. Jaroslav Pokorný

Acknowledgments First of all I would like to thank my supervisor Prof. Jaroslav Pokorný for his unbelievable patience, encouragement and invaluable advice on the text of the thesis. I am indebted to Prof. Ingemar J. Cox for receiving me at UCL, his impressive energy, research guidance, and spot on insight. Seoung-Taek Park was very kind to host me at Yahoo! Research in Burbank during the three months in the summer of 2007 and we had many interesting and intensive discussions. My collaborators – Tobias Escher, Martin Bög, Helen Margetts, C. Lee Giles, Isaac Councill, and Hui Han – all taught me during our interaction more than they can imagine. I had great fun working with every one of them.

Many other people helped during my research. Tom Rutter was extremely helpful during the execution of the website user study at the ELSE laboratory, Neil Marjoram and Rich Hutchinson both went an extra mile when supporting my work on computer systems at UCL's Adastral Park campus.

Several datasets have been used in this work. Yahoo! research provided datasets for social collaborative filtering experiments, Michael Ley provided the DBLP dataset and kindly discussed all aspects of DBLP, the digital library he runs, by email and phone. C. Lee Giles and Isaac G. Councill provided the CiteSeer data and help with my understanding of this data.

This work would not be possible without the financial support of Dagmar and Vaclav Havel's Foundation Vize 97, Mobility fund of Charles University, Bernard Bolzano foundation, Yahoo! Research, Cambridge-MIT Institute and National programme of research (Information society project 1ET100300419).

Sorry to all my friends and family who heard from me and saw me so rarely during this time. Thanks for the support and understanding.

Many thanks go to Jorge Cham for doing that comic strip about me, to Jan Pechanec for feedback on this text and the most special thanks go to my fiancee Isa(belle) for bearing me during the writing up, feeding me, and actually making this thesis happen by application of the necessary pressure.

I certify that I have written my thesis by myself and only using the references cited. I agree with lending my thesis.

Cambridge, May 14, 2007

. . . . . .

Abstract

Title: Quantitative analysis of networked environments to improve performance of systems
Author: Václav Petříček
Department: Department of Software Engineering
Supervisor: Prof. Jaroslav Pokorný
Supervisor's e-mail address: jaroslav.pokorny@mff.cuni.cz
Abstract :
In this thesis we encounter networks in three contexts i) as the citation networks between documents in citation databases CiteSeer and DBLP, ii) as the structure of e-government websites that is navigated by users and iii) as the social network of users of a photo-sharing site Flickr and a social networking site Yahoo!360. We study the properties of networks present in real datasets, what are the effects of their structure and how this structure can be exploited.

We analyze the citation networks between computer science publications and compare them to those described in Physics community. We also demonstrate the bias of citation databases collected autonomously and present mathematical models of this bias. We then analyze the link structure of three websites extracted by exhaustive crawls. We perform a user study with 134 participants on these websites in an lab. We discuss the structure of the link networks and the performance of subjects in locating information on these websites. We finally exploit the knowledge of users' social network to provide higher quality recommendations than current collaborative filtering techniques and demonstrate the performance benefit on two real datasets.

Keywords: Networks, citation, collaborative filtering, recommenders system, fusion, algorithm

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

With the advent of the Web many traditional applications have found their way online. The automated transaction handling, global exposure and resulting increase in number of users started an enormous data explosion. Not only there is more data available now than anytime before but also qualitatively new types of data can and are being collected.

Networks[1] are an integral part of much of the data available. The information published online is held together by the network structure formed by hyperlinks connecting individual documents and allowing us, the surfers, to navigate towards related content. Search engines, research institutions and non-profit organization all crawl the Web, collecting ever growing snapshots of the Web. Another type of networks is being extracted from scientific papers which are referring to each other by a web of citation links. On social networking sites, users create their profiles, socialize and link to their friends. This network of social links presents an unprecedented amount of information that was hard to get in the past. Researchers needed to perform surveys and interviews, either online or in person, to collect a small subset of social, economic and other networks. This data is now readily available to operators of online sites and applications on a scale previously unimaginable.

Information systems and other systems in general are affected or affect the structure of various networks. Search engines exploit the link structure between web pages to provide ranking, this structure links also affects surfing habits of users and the information they are exposed to, marketing campaigns rely on social networks to facilitate product adoption, computer networks collapse as a result of attacks on malicious nodes, the way we choose our friends reflects our own taste and experience etc. It is necessary to un-

[1] Here we do not mean computer networks in the narrow sense of interconnected computers, but general networks of interconnected entities, be it documents, machines, or people.

derstand networks and the effects of their properties as it may lead to better interpretation of data gathered in networks, better design of systems relying on networks and better algorithms exploiting these networks.

Today there are several huge datasets containing data on some of these networks: the links between millions of web pages; we are able to map the global topology of the Internet at the level of autonomous systems; large citation databases of scientific articles in many scientific disciplines are available; and many social networking sites expose social networks of millions of people. The large scale of these datasets makes them well suited for quantitative analysis.

Network analysis is an application of graph theory [91] and statistics to the study of real and artificial networks. Network analysis has been developing relatively independently within disciplines as varied as Physics [154], Mathematics [65], Social network analysis [222], and Computer science [26]. Each of these disciplines work with networks that have fundamentally different semantics, yet some methods are applicable in all of the disciplines.

Networks are not all the same. The differences of networks may be well illustrated on the three networks we study in this thesis: each of them the citation network, the website link structure and the social network consist of interconnected elements. The semantic of a link is very different though. In citation analysis links are most commonly interpreted as endorsements, inside websites as a navigation path and social links may express friendship for example.

The availability of some of the data at this scale raises privacy and security issues [14]. Knowledge of the Internet topology may be used for targeted attacks for example. Teenagers and children have scared their parents by the amount of sensitive information they are willing to publish about themselves on social networking sites such as MySpace. The knowledge of the global social network may be abused to discriminate against or target minorities and other vulnerable groups. The threats perceived have spurred intense discussions and legislative efforts. Still it is highly likely that network data including social network data will always be available. At least transaction mediators will be able and motivated to analyze such data. If the insights of their analysis are used carefully, customers and users will benefit as well. The analysis of Web structure data, which is less sensitive and has been available for longer time, has already brought tremendous progress in information retrieval, data clustering, bibliometrics and many other areas.

A serious negative aspect of the data and information availability in today's society is information overload. Users suffer from having to make too many decisions and to review and absorb overwhelming amount of information. This results in sub-optimal user performance and choices, given

the amount of information actually available, frustration and low customer satisfaction. Additional information and bigger choice can actually have a negative impact in retail. Users facing too many options are afraid to make a bad choice and do not buy anything.

A successful approach to tackling information overload have been recommender systems which help users providing a list of selected items that they are likely to enjoy. Well known examples of recommender systems are Amazon's "Customers who bought this item also bought ..." feature[2]. Other popular sites using recommender systems are MovieLens[3] and NetFlix[4] movie recommenders and Last.fm[5] and Pandora[6] music radios. The term recommender systems has been coined by Resnick and Varian in the special issue of Communications of the ACM on recommender systems [186]. The most prevalent recommender systems are implemented using collaborative filtering. Collaborative filtering uses the preferences of many users to filter items that the user may like. It can for example collect purchase history of users, their surfing habits, measure the time they spend reading articles, or solicit explicit preferences by asking users to rate items on a continuous or discrete scale. The basic idea behind recommender collaborative filtering systems is that users who agreed in the past are likely to agree in the future. There are other ways to implement recommender systems, such as rule based recommenders relying on human experts for the rule set construction, or knowledge based recommenders using ontologies for example. Collaborative filtering, though, is by far the most successful and popular way to implement recommender systems. Recommender systems may help users discover books, movies, and other items they never heard of and which were written by authors they never thought existed. When recommender systems work well, both the providers and the users benefit.

Even though many different networks have been studied in different contexts (we present a review of this literature in Section 4.1), there are still areas unexplored where network analysis can bring interesting insights. Despite the amount of work on recommender systems and several applications of network analysis to recommenders there is an unexploited opportunity to combine these two to improve provide better recommendations.

Study of network structures can be informative to improve performance of systems - be it websites, scientific citation databases, or recommender systems. These improvements in performance can lead to i) more accurate

[2] http://amazon.com
[3] http://movielens.umn.edu
[4] http://netflix.com
[5] http://www.last.fm/
[6] http://pandora.com

assessment of publication records in bibliometrics; ii) better website design and user experience evaluation and iii) higher user loyalty and improved monetization of services employing our recommender system.

The goal of this thesis is to analyze networked environments and their properties and look for network properties that influence the performance of systems. We would like to find properties that can be possibly exploited to improve the performance of existing systems. We will study three different types of network environments i) citations between scientific documents, ii) pages connected by hyperlinks within a website and iii) social network of friends and acquaintances. Compare, evaluate and attempt to improve performance through understanding of the related networks.

In this thesis we present three studies which contribute to the vast body of work on citation analysis, hyperlink analysis, social network analysis and recommender systems,

- We compare the citation distribution in two large computer science citation databases. We contrast the distributions with previous results published for Physics community in [134]. For the first time we also compare the acquisition methods and study the bias introduced by the self-selected nature of user submissions and crawling bias towards papers with higher number of authors.

- We study the behavior of 134 users during a controlled navigational study in an experimental computer lab where they were instructed to perform tasks on three comparable websites. We report the structural properties of these websites and discuss the effects of these properties on the navigability of the websites. Our website structure analysis is novel in that it focuses on previously neglected class of websites - the e-government websites (foreign offices in particular). Due to the financial and technical requirements there have been so far very few user based studies where users could be observed in a controlled environment.

- We use the knowledge of users' social network to implement two types of network-informed recommender systems: i) a pure social recommender and ii) combined social-collaborative filtering recommender. We perform quantitative evaluation of these new algorithms on real world datasets from Yahoo!360 social networking site and Flickr photo sharing site. We show how the knowledge of the social network data improves recommendations – especially to users with little preference information available. Our use of social network for item recommendations and especially the combination of social network information with traditional collaborative filtering recommenders is following a previously

unexplored direction. This research has been made possible thanks to the recent availability of large datasets of network and preference data. Unlike previous hybrid recommender systems this approach does not use item content or user demographic data. Instead a qualitatively new type of data associated with users is exploited – real social network.

To our best knowledge these are the main contributions of this thesis which haven't been presented before.

The thesis is organized as follows. We first provide a description of formalisms, concepts and notation for network analysis in Chapter 2 and for recommender systems in Chapter 3. We then review related previous work in Chapter 4. The literature review is organized by area into three parts – 1) network analysis and 2) recommender systems and 3) applications of network analysis in recommender systems. Chapters 5, 6 and 7 present our main contributions, in the different but related areas of citation analysis, website analysis and recommender systems. In Chapter 5 we perform analysis and comparison of two computer science citation databases and confront the observations with results published for physics. We also analyze the bias introduced by self-selection and crawling as acquisition methods. Next we perform an evaluation of e-government websites in Chapter 6. We perform a user based study of the navigability of three e-government websites together with the analysis of their link structure. Finally in Chapter 7 we describe a set of network informed recommender systems for item recommendation which exploit the implicit user similarity signaled in users' social ties. We present a quantitative evaluation of the performance of these algorithms on Flickr and Yahoo!360 datasets. Chapter 8 provides a summary of our contributions and we discuss possible directions of our future work. Some additional detailed information has been included in Appendices.

Chapter 2

Network analysis background

In this chapter we provide background on network analysis as related to our work. This is not aiming to be an exhaustive list of all the concepts ever introduced in network analysis – rather a summary of definitions relevant to this thesis is presented here.

We study various networks including websites' structure, citation networks, and social networks. Adopting the terminology of Graph Theory [91], we refer to web pages, papers, users as *vertices* and to hyperlinks, citations, social ties and other relations between vertices as *edges*. Vertices are connected by edges to form a *graph* (network). As we are dealing with often asymmetric relations (citations for example) we consider only directed graphs with directed edges.

We first introduce notation, then define several vertex, edge and network properties and finally we describe a few network algorithms that will be useful later.

2.1 General notation

We use several norms. The cardinality, number of elements, of a set $\mathcal{V}$ is denoted as $|\mathcal{V}|$. For vectors and matrices the L_1 norm corresponds to the sum of its elements: $\|\mathcal{E}\|_1 = \sum_{e_{i,j} \in \mathcal{E}} e_{i,j}$.

2.2 Network graph

Network graphs can represent many relationships: a paper v citing paper u, a page pointing to another one, or a user v listing user u as his friend. We treat a network as directed graph denoted $\mathcal{G}$ or $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set

of vertices and $\mathcal{E}$ set of edges. $\mathcal{E}$ is represented by a square binary matrix where an entry of this matrix, $e_{v,u}$, is one if and only if there is a directed edge from vertex v to vertex u and it is zero otherwise. Note that, due to the directed nature of the graph (if there is a link from one page to an other it does not necessarily mean there is a link back), $e_{v,u} = 1$ does not imply that $e_{u,v} = 1$. We assume that the network graph is free of self-ties (edges originating and ending in the same vertex): $e_{v,v} = 0$ for all v. Due to the fact that the connection matrix $\mathcal{E}$ is binary, multi-edges are treated as a single edges. We measure the density of the matrix $\mathcal{E}$ by computing:

$$\omega(\mathcal{E}) = \frac{\|E\|_1}{|\mathcal{V}|^2} \tag{2.1}$$

The density $\omega(\mathcal{E})$ is the fraction of number of non-zero elements in matrix $\mathcal{E}$ (number of observed edges) divided by the total number of matrix elements (number of all possible edges). If necessary we refer to vertices of network $\mathcal{G}$ as $\mathcal{V}_\mathcal{G}$ and to edges $E_\mathcal{G}$ to indicate the network in question. We denote by $n_\mathcal{V}$ the number of vertices and by $n_\mathcal{E}$ the number of edges ($n_\mathcal{V} = \|\mathcal{V}\|$ and $n_\mathcal{E} = \|\mathcal{E}\|_1$).

Path from vertex v to u is a sequence of vertices $u_0 \dots u_l$ such that $u_0 = v$ and $u_l = u$ and $\forall_{i=(0..l-1)} : e_{u_i,u_i+1} = 1$. l is the length of the path. We are considering directed paths only. *Strongly connected component* is a subgraph $\mathcal{G}' \subset \mathcal{G}$ where there exists a directed path between all pairs of vertices in $\mathcal{G}'$. The word strong means this directed path exists from v to u as well as from u to v. *Distance* $d(u, v)$ is the length of shortest path between v and u. $d(u, v) = \infty$ if there is no path from u to v. *Degree* of a vertex v is denoted $\rho(v)$ and represents the number of edges incidental with this vertex. *In-degree* $\rho_{in}(v)$ is the number of edges coming in vertex v. *Out-degree* $\rho_{out}(v)$ is the number of edges going out of v. Naturally $\rho(v) = \rho_{in}(v) + \rho_{out}(v)$. *Reachability of vertex* v, denoted $\gamma(\mathcal{G}, v, \tau_d)$, is the proportion of the network vertices u for which $d(v, u) < \tau_d$. This is, for example, the proportion of the network accessible from vertex v in τ_d or less clicks.

2.3 Edge interpretation

Many relations can be modeled as a network and from the variety of situations comes a variety of interpretations of a link. Reasons for creation of a link may differ for each two vertices but generally there is a most common cause or reason that determines appropriate interpretation. This interpretation is very important as it determines the conclusions we may draw from network properties. Broadly, interpretations of a link may be

- endorsement
- communication
- navigation
- similarity

If links are interpreted as endorsements, we may look for the most authoritative vertices. If link is a means of communication or navigation we could discover bottlenecks in the network, single points of failure, etc. Links signaling similarity can reveal clusters of vertices/users with similar taste. In Section 4.1 we present a literature review of studies that focus on different link interpretation.

2.4 Attribute data

Graph vertices often have additional properties – scientific papers have names of authors associated with them, web pages may be annotated by keywords or tags, users list items they like, etc.

In computer science such properties are represented as attributes. An attribute of vertices is a function which assigns a value to each vertex. For example an attribute "year" assigns to each document an integer representing its publication year. An attribute "favorites" assigns to each user a list of his her favorite items or attribute "author names" assigns a list of authors to each paper.

In this thesis we will be studying and working with simple numeric attributes and list attributes. We will treat both types in similar way as numeric attributes may be seen as assigning a list of length one to each vertex. Attributes will be represented by a a triple $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$, where $\mathcal{V}$ is the same set of vertices as in the graph above. $\mathcal{I}$ is the set of all possible list elements assigned by the attribute – all items, years, or all author names for example. $\mathcal{F}$ then represents the relationships between vertices $\mathcal{V}$ and the elements of $\mathcal{I}$.

Lets see a few examples. Publication year data for documents may be represented as $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$, where $\mathcal{V}$ = documents, $\mathcal{I}$ = 1900..2007, and $\mathcal{F}$ is a matrix where for each $v \in \mathcal{V}$ and $i \in \mathcal{I}$ $F_{v,i} = 1$ if and only if document v has been published in year i. Users' favorite lists may be represented as $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$, where $\mathcal{V}$ = users, $\mathcal{I} = items$, and $\mathcal{F}$ is a matrix where for each $v \in \mathcal{V}$ and $i \in \mathcal{I}$ $F_{v,i} = 1$ if and only if item i is a favorite of user v. The type of attribute will be obvious from the context and we will make the distinction

where necessary. Finally, authorship data may be represented as $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$, where $\mathcal{V}$ = papers, $\mathcal{I} = authors$, and $\mathcal{F}$ is a matrix where for each $v \in \mathcal{V}$ and $i \in \mathcal{I}$ $F_{v,i} = 1$ if and only if author i has written paper v.

$\mathcal{F}$ is a sparse rectangular binary matrix.

We denote the number of possible items and number of user-item co-occurrences as $n_{\mathcal{I}} = |\mathcal{I}|$ and $n_{\mathcal{F}} = |\mathcal{F}|_1$ respectively. We also refer to the set of items associated with vertex v as $\mathcal{F}_v : \{v : f_{v,i} = 1\}$. $\mathcal{F}_v$ may therefore represent a set of authors of a paper, set of words on a page or set of user's favorite pictures. We measure the density of the attribute matrix $\mathcal{F}$ by computing the fraction of observed co-occurrences (non-zero elements) out of all possible co-occurrences:

$$\omega(\mathcal{F}) = \frac{\|\mathcal{F}\|_1}{|\mathcal{V}||\mathcal{I}| = \frac{n_{\mathcal{F}}}{n_{\mathcal{V}} \cdot \mathcal{I}}} \tag{2.2}$$

where $|\mathcal{V}|$ and $|\mathcal{I}|$ are the dimensions of matrix $\mathcal{F}$.

This framework can be easily extended to allow for non-binary relations but binary relations will be sufficient for results presented in this thesis.

2.5 Network properties

Taking one step back and looking at the network as a whole we can identify various structural properties some of which we introduce here.

Bow-tie structure has been described and measured by Broder et al. [29] for the Web but similar structure may be found in any directed network. Example of a bow-tie structure is in Figure 2.1. The individual parts are *LSCC*, *IN*, *OUT*, *TUBEs*, *TENDRILs* and *DISCONNECTED* components. *LSCC* is the largest strongly connected component. There is a directed path between any two vertices in *LSCC*. *OUT* contains all vertices for which there is a path from the *LSCC* but which are not part of *LSCC*. There exists a directed path from each vertex in *LSCC* to each vertex in *OUT* but not the other way around. *IN* component contains all vertices such that it is possible to find a path from anywhere in *IN* to the *LSCC* (and therefore to *OUT*) but not the contrary. *TUBE* components connect *IN* to *OUT* in a one-way fashion – it is possible to reach *OUT* from *IN* but not vice versa. *TENDRILs* are, simply, the remaining dangling bits – the remaining nodes reachable from *IN* and nodes from which there is a path to *OUT* but they do not belong to any other component. *DISCONNECTED* components are separated parts of the network that are not connected to any other component. The parts form a structure reminiscent of a shape of a bow-tie. The size of the individual components affects the overall navigability of the network.

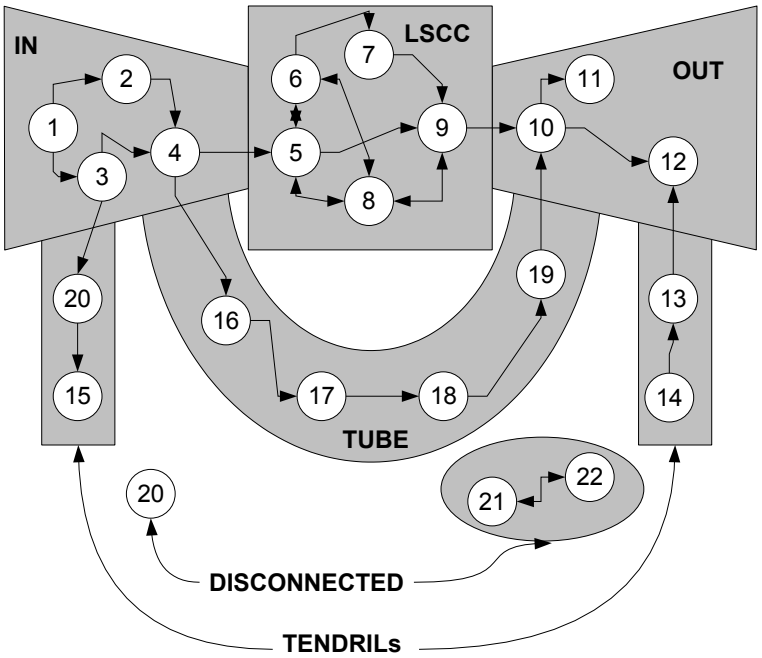


Figure 2.1: Illustration of the graph theoretic concepts. There is a path from vertex (1) to vertex (10) which uses vertices (2),(4),(5) and (9). There is no path from vertex (12) to vertex (4). The distance between (4) and (10) is 3. The degree of vertex (9) is $\rho(9) = 5$, the in-degree $\rho_{in}(9) = 3$ and out-degree $\rho_{out}(9) = 2$. The shaded areas divide graph according to "bow-tie" structure (Section 2.5) according to navigability between the different parts. The resulting parts resemble a bow-tie. Note there are several strongly connected components but (5)(6)(7)(8)(9) is the largest.

Reachability of network, $\Gamma(\mathcal{G}), \tau_d)$ is the average reachability of individual vertices in the network: $\Gamma(\mathcal{G}, \tau_d) = \frac{\sum_{v \in \mathcal{V}_\mathcal{G}} \gamma(\mathcal{G}, v, \tau_d)}{|\mathcal{V}|}$. *Diameter* $D(\mathcal{G})$, is the longest of all shortest paths between any two vertices in the network. *Directed average distance*, $\bar{d}(\mathcal{G})$, is average length of shortest paths between all pairs of vertices in the graph $\mathcal{G}$ where unreachable pairs are ignored. $\bar{d}(\mathcal{G}) = \frac{\sum_{u,v \in \mathcal{V}: d(u,v) \neq \infty} (d(u,v)}{\sum_{u,v \in \mathcal{V}: d(u,v) \neq \infty} 1}$ *Percentage of unreachable pairs* percentage of pairs of vertices, (v, u) where there does not exist a directed path from the vertex v to the vertex u. *Average degree* $\bar{\rho}$ – average number of links (incoming and outgoing) per vertex how dense a graph is but contrary to density of a network it is independent of size. *Degree distribution* is a probability distribution of degrees in the network [65]. Many natural networks exhibit a power-law distribution which exhibits itself as a straight line on a double

logarithmic plot.

Homophily is the tendency of vertices to link to other similar vertices. Homophily of the network with respect to property i may be measured by the $\chi(i)$ coefficient:

$$\chi(i) = \frac{N \cdot \sum_{(u,v)\in E(\mathcal{G}):r_{u,i}=1,r_{v,i}=1} 1}{\sum_{(u,v)\in E(\mathcal{G}):r_{u,i}=1} 1 \cdot \sum_{u:r_{u,i}=1} 1} \tag{2.3}$$

This coefficient quantifies the tendency of vertices with property i to associate with other vertices with feature i. We call an edge i-homophilous if it connects two vertices both of who have property i. Coefficient $\chi(i) = 1$ means that the social network structure and the distribution of property i are independent, or that the fraction of i-homophilous edges is the same as if the vertices were choosing their neighbors independently of i.

2.6 Network algorithms

In graph theory, breadth-first search (BFS) is an uninformed graph search algorithm that begins at the root node and explores all the neighboring nodes. Then for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until termination criterion is met. A FIFO queue is generally used for nodes to be explored.

Depth-first search (DFS) is again an uninformed search that progresses by expanding the first neighbor and thus going deeper and deeper until a termination criterion is met, or until it hits a node that has no unexplored neighbors. Then the search backtracks, returning to the most recent node it hadn't finished exploring. In a non-recursive implementation, a LIFO stack is used.

Focused search is and informed graph search algorithm which uses a priority queue. This queue is updated each time a new vertex is being explored. We use a queue sorted by number of edges that are originating from vertices already explored. Each time the vertex with the highest in-degree is explored. This results in the algorithm first exploring dense subgraphs before venturing further in the network. A pseudcode of the algorithm follows.

Algorithm 1 FS($\mathcal{G}$,s)

```
1: Q[s] = 1
2: while v = argmax_v Q[v] do
3:    Q[v] =⊥
4:    for all u : e_{v,u} == 1 do
5:       if Q[u] ≠⊥ then
6:          Q[u]++
7:       end if
8:    end for
9: end while
```

2.7 Summary

We introduced basic concepts of network analysis which we use further in this thesis. In the rest of the thesis we investigate several different datasets containing network data. In the next chapter we look at the differences between citation databases. We then study in Chapter 6 the structure of websites and navigation of these websites by real users. Finally we demonstrate how homophily present in social network can be exploited to improve performance of recommender systems.

Chapter 3

Recommender systems background

In this chapter we introduce concepts of recommender systems using the same framework as we used in the previous chapter. We also describe algorithms that we will use later in this thesis – particularly in Chapter 7.

Recommender systems try to address the recommendation problem: "Given the history of user's favorite items, predict additional, previously unseen, items the user will enjoy." One of the most successful approaches to solving this problem has been collaborative filtering. Collaborative filtering utilizes the history of many users to find similar users and items that serve as a basis for the recommendations. We first formalize recommender systems and then we will describe a non-personalized algorithm POP and two well-known collaborative filtering recommenders – the user-user and item-item algorithms.

3.1 Notation

Recommender systems make use of data about users – their preference history – or simply attributes. As in the previous chapter, the *attribute data* is in the form $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$, where $\mathcal{V}$ denotes users, $\mathcal{I}$ items and matrix $\mathcal{F}$ denotes a user-item matrix of favorite items. Rows and columns of the matrix represent users and items respectively. An entry f_{ui} of the matrix $\mathcal{F}$ is equal to one if the item i is a favorite item of user u, otherwise, f_{ui} is zero. We also sometimes refer to f_{ui} as a *rating* by user u, of item i. The term rating comes from collaborative filtering where ratings may have integer or even real values. Note that in our case the ratings f_{ui} are binary. For a given user, u, we also refer to those items for which f_{ui} is one as *known items* or *known favorites*. Row corresponding to user u and column corresponding to

item i of matrix $\mathcal{F}$ are denoted as $f_{u,\star}$ and $f_{\star,i}$ respectively. We refer to the user for whom we are generating the recommendations as *active user* or user a.

The recommendation problem may be then formalized as: "Given users' history $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$ and active user a, determine N items $\in \mathcal{I} \setminus \mathcal{F}_a$ that are most likely to be enjoyed by user a."

3.2 Recommender Algorithms

Recommender algorithms can be seen as consisting of two parts – *score calculation* and *recommendation generation*. During the first stage, for active user a, a score w_{ai} is computed for each item reflecting how likely a user a is to select an item i as her favorite. This score is not necessarily a probability. Note that we focus on the top-N recommendation problem rather than the *prediction problem*. In the latter, accurate prediction of a user's preference for a specific item is important. Thus, the score of each item is often interpreted as a predicted rating or a probability that an item will be consumed by a specific user. Performance of algorithms is then measured by the average difference between actual and predicted ratings. However, where recommender systems are used to support user decisions, e.g. "what film should I order?", users do not require a precise rating prediction for all items, provided relevant items are displayed. Thus, in our work, the scores are only used to find the top-N items for recommendation. The *recommendation generation* step is identical for all algorithms. Algorithms differ in the score calculation only. After scores of all items are calculated for a given user, items are sorted by score in descending order and items already consumed by the user are removed. Finally, the top-N items with the highest scores are selected for recommendation.

3.2.1 Non-personalized baseline (POP)

The POP algorithm recommends items based on their global popularity (hence POP) across all users. The popularity of an item is measured by the number of users who have selected it as a favorite. The algorithm presents a user, a, with the N most popular items, excluding those items which the user has previously selected. Algorithm 2 presents pseudocode of the score calculation step for active user a. Line 2 counts the number of non-zero elements in column corresponding to item i.

The item popularities needed may be easily precomputed and the resulting recommendations are very fast. Memory requirements are also very low

Algorithm 2 POP(a)

1: **for all** $i \in \mathcal{I}$ **do**
2: $\quad w_{ai} \leftarrow \|f_{\star,i}\|_1$
3: **end for**
4: **return** w_a

as we do not need to keep the whole matrix in memory. Popularities may be incrementally updated as users provide new ratings and the ranking of items is generally quite stable.

3.2.2 User-user algorithm (UU)

The user-user algorithm [96] utilizes the k-most similar users to the active user to make recommendations. The rationale behind this is that users who have similar taste based on their past preferences are likely to agree in the future. Therefore items popular between users similar to active user are likely to be enjoyed by active user too. The user-user algorithm first calculates the similarity, based on cosine distance [189], between user a and all other users v, i.e.

$$sim_U(a, v) = \frac{\sum_i (f_{ai} \cdot f_{vi})}{\sqrt{\sum_i (f_{ai})^2 \cdot \sum_i (f_{vi})^2}} \tag{3.1}$$

Then for user, a, the algorithm identifies her k most similar other users, K_a^k. This set is then used to calculate the weighted popularity of each item, i, for user, a:

$$w_{ai} = \sum_{v \in K_a^k} sim_U(a, v) \cdot f_{vi} \tag{3.2}$$

where f_{vi} is either 0 or 1. Finally, the items are sorted in decreasing order of score, w_{ai}, excluding items marked by the user a as favorites, and the top-N items are recommended to user, a.

The similarity computation utilizes a threshold on the minimum number of items that two users have in common. If this threshold is not exceeded, then the similarity between the two users is set to zero. Thus, a user-user algorithm has two parameters, the number of nearest neighbors, k, and a minimum overlap threshold, τ. A user-user algorithm with specific parameter values is denoted UU(60,1), corresponding to $k = 60$ and $\tau = 1$. Algorithm 3 presents pseudocode of the score calculation step of UU.

There is a disadvantage to UU approach – the similarity between users cannot be precomputed without sacrificing quality of recommendations. The

Algorithm 3 UU(a, k, τ)

1: **for** $u \in K_a^k$ **do**
2: $\quad w_{ai} \leftarrow w_{ai} + sim_U(a, u) \cdot f_{ui}$
3: **end for**
4: **return** w

similarity between two users can change dramatically with one additional rating, especially when one or both of them provided little ratings before. When precomputing the similarities, the system would not react to additional ratings entered by the active user and this could discourage users from providing ratings. The next algorithm has been motivated by this disadvantage of UU algorithm.

3.2.3 Item-item algorithm (II)

The approach of item-item algorithm is in a way dual to that of user-user algorithm. Instead of looking for users similar to the active user, II finds k most similar items to each of the items that active user liked in the past. Hopefully, items often bought together with the ones acquired already by active user form a good basis for recommendations to this user.

The item-item algorithm [59] first normalizes each row of the matrix $\mathcal{F}$ such that $\sum_u f'_{ui} = 1$ where $f'_{ui} = f_{ui} / \sum_i f_{ui}$. The normalization decreases the influence of highly active users who have rated many items.[1] Item similarities are then measured by cosine distance. For each item, j, in the user's favorite set, $\mathcal{F}_u$, we compute

$$sim_I(i, j) = \frac{\sum_u (f'_{ui} \cdot f'_{uj})}{\sqrt{\sum_u (f'_{ui})^2 \cdot \sum_i (f'_{uj})^2}} \tag{3.3}$$

and identify the set, K_j^k, of k most similar items to j.

The score of an item, i, is then calculated as a sum of similarities between i and the favorite items, $\mathcal{F}_u$, of user, u:

$$w_{ui} = \sum_{j \in \mathcal{F}_u} sim_I(i, j) \cdot \delta_{ij} \tag{3.4}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i \in K_j^k \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

[1] Such normalization has no effect on UU.

is an indicator variable that is one if item, i, is one of the k-most similar items to j, and zero otherwise. Finally, items are sorted in decreasing order of score, excluding items in the user u's favorites, and the top-N items are recommended.

The item-item algorithm utilizes a threshold, τ, on the number of users that two items have in common. If this threshold is not exceeded, then the similarity between the two items is set to zero. Thus, an item-item algorithm has two parameters, the number of similar items, k, and a minimum overlap, τ. An item-item algorithm with specific parameter values is denoted II(50,2), corresponding to $k = 50$ and $\tau = 2$. Algorithm 4 presents pseudocode of the score computation step of algorithm II.

Algorithm 4 II(a, k, τ)

```
for j ∈ F_a do
    for i ∈ K_j^k do
        w_ai ← w_ai + sim_I(j, i)
    end for
end for
return w_a
```

3.3 Summary

We have described three recommender systems. We use these recommender systems in Chapter 7 to build more advanced recommender systems informed by network structure. We also use them for comparison in quantitative evaluation. In the next two chapters we are going to provide review of related previous work in the fields of network analysis and recommender systems.

Chapter 4

Previous work

In this chapter we review previous related work in the areas where there are contributions of this thesis. We also summarize the main new contributions of the thesis and the main differences with previous studies.

The contributions of this thesis have two aspects i) analysis of datasets containing network data and ii) use of the network data to implement a recommender system. We organize this chapter into three sections i) a review of previous network analysis work (Section 4.1), ii) review of previous recommender systems work (Section 4.2) and finally iii) previous applications of network analysis in the recommender systems research community (4.3).

4.1 Previous network analysis work

Network analysis is the application of graph theory [91] to the analysis of real world networks or their models. Networks of different kind have been analyzed in many disciplines be it social or natural sciences. These disciplines differ in the type of networks they study, tools they use and objectives they are trying to achieve. In this thesis we present our analysis of hyperlink structure of websites, citation networks and then social networks. We therefore review here the related previous work in social network analysis (Section 4.1.1), scientometrics (Section 4.1.2), and networks in computer science (Section 4.1.3), which are the most relevant to our study.

4.1.1 Social network analysis

Social network analysis studies the patterns of interconnections between vertices and the mutual interaction of the existing edges and vertex attributes. Barnes [13] is credited with coining the term *social network analysis* although

the origins of the discipline date back into 1930's. Social network analysis is a technique often used in social sciences such as sociology, anthropology, psychology, epidemiology and many other disciplines. Its notation is historically different from the one we use in this thesis and which we described in the Chapter 2. In social network analysis the *vertices* are called *actors* and they usually represent people, organizations or other agents, *network data* is referred to as *relational data* as it represents the relations between actors and instead of *edges* researchers talk about *dyads*. There is a difference between a dyad and edge though - there are three types of dyad i) a missing edge ii) single directed edge and iii) two reciprocal directed edges.

Social network analysis has developed significantly. First studies were mainly descriptive, computing network statistics such as density, triad census, triangle coefficient, degree distribution, and many others [157, 222]. This approach allows the comparison of different networks or instances of the same network over time. Frequently observed effects in the networks are reciprocity (vertices tend to connect to others who link to them), transitivity (if v_1 links to v_2 and v_2 links to v_3 then v_1 links to v_3 too. An attribute related effect has been dubbed homophily and has been observed in many real life scenarios. In principle a network may be homophilous or heterophilous. In homophilous network vertices with similar attributes tend to link to each other as expressed in the adage "birds of feather flock together". Many networks such as friendship networks [5], advice networks or co-authorship networks exhibit homophily. An example of a heterophilous network is a network of sexual contacts with respect to gender. McPherson et al. [146] presented a comprehensive review of homophily results. Descriptive social network analysis is the oldest and most developed area of social network analysis.

Recently social network analysis has been popularized by books such as [223] and [12]. Social network analysis has been frequently used in organizations to understand the workings of companies and many consultants have established themselves.[1]

Qualitative studies are common in some disciplines – anthropology for example. These studies generally deal with smaller networks and rely heavily on the judgment of the research or during interpretation of the character of individual links and reasons for their creation. Such studies are of limited relevance for us as we adopt a more objective quantitative approach.

Orthogonal to our research, visualization techniques have been developed during exploratory analyzes of social networks. Several computer programs and libraries have been developed and are available online[2]. Statistical

[1] http://www.orgnet.org

[2] http://www.insna.org/INSNA/soft_inf.html

modeling techniques have been developed to address the problem which descriptive social network analysis has with differentiating between competing explanations. The statistical models allow estimation of parameters and hypothesis testing. They answer questions such as "do people make friends with their direct coworkers?" or "are two networks similar?". Wassermann et al. [222] and Carrington et al. [40] provide a more detailed description of the various modeling approaches discussed below.

The simplest models in use have been introduced by Erdos and Renyi [65] and represent a class of random graphs. In random graphs the probability of an edge is equal for each possible edge. Random graphs result in Poisson distribution of vertex degree which is in contrast with empirical observations. These models are generally used only as a null hypothesis to test for statistical significance of non-random effects in link formation. More complex Markov models [70] account for the frequencies of network artifacts such as single edges, stars and triangles. Holland et al. [102] proposed a model, *p1*, that assumes dyad independence and accounts for reciprocity, in-degree and out-degree of vertices and density of network. Model *p2* [218] added the effect of vertex attributes. Under this model the edges are individually independent given the values of attributes of vertices they connect. A generalization of Markov, *p1*, *p2* and other related models has been recently proposed by Snijders et al. [204]. Their generalization included the frequencies of other artifacts such as k-triangles, k-paths, and k-out stars in the model.

The models mentioned so far modeled the effect of vertex attributes on edge creation. This effect is called *(social) selection* [40]. The reverse mechanism, in which the network structure and the particular nodes to who a vertex is connected affect the attributes of the node, is called *influence* or *contagion* [216] depending on context. While *social selection* reflects how people select their friends, *influence* and *contagion* reflect, for example, how a person's taste in music is influenced by their friends or how a disease or innovation spreads in the population [188], or how peer pressure of friends who smoke increases a chance that a teenager will start smoking [217]. Contagion modeling has caught the interest of marketers, who traditionally use well established global models that ignore the structure of networks and predict that adoption of new products roughly follows a logistic curve [187]. The relationship of network structure and vertex attributes may be also tested by autocorrelation [62]. In autocorrelation test the attribute data is used to compute a similarity matrix, which is then correlated with the network incidence matrix using quadratic assignment procedure (QAP) [107]. Unfortunately the exact QAP is NP-complete and therefore Monte-Carlo simulation has to be used for larger networks.

Social network studies are related to our work on exploiting social net-

work for item recommendations. We take advantage of the homophily signaled by the social connections. Unfortunately, the techniques used in social network analysis are focused on hypothesis testing and ignore the computational complexity of the procedures. This makes them ill-suited for large scale item recommendations in their current form.

4.1.2 Scientometrics

Scientometrics [3] a part of of bibliometrics – the study of published texts and their relationships. Citation analysis studies the citation links between these texts. In scientometrics, citation is generally interpreted as endorsement and used for ranking, or interpreted as a sign of similarity and used to find related documents Citation analysis has been pioneered in bibliometrics in the context of scholarly articles.

Broadly, prior citation analysis has examined a wide variety of factors including (i) the distribution of citation rates [184, 134, 36, 126], (ii) the variation in the distribution of citation rates across research fields and geographical regions [134, 117], (iii) the geographic distribution of highly cited scientists [16, 17] (iv) various indicators of the scientific performance of countries [143] (v) citation biases and mis-citations [122, 123, 202] (vi) collaboration networks [153] (vii) distribution of references in papers [220], and (viii) visualization and navigation [121, 54].

There are now many public citation databases available. The database of BibTeX bibliographies [55], Networked Computer Science Technical Reference Library [158], CiteSeer [132] - an autonomous indexing engine and DBLP, a citation database originally covering Database research and Linear Programming but today much more. [56] Originally independent libraries Compuscience [50] and CoRR [52] became part of Arxiv [10] – a database popular especially with physicists. Another library specializing exclusively in High Energy Physics is [206]. There also exist proprietary databases most notably the Scientific citation index [196] or [197].

In addition to collection problem there is also the issue of efficiently presenting the content and visualizing information derived from the accumulated papers. Klink et al. [121] described a DBLP interface for browsing publications by authors, coauthors, dates, and other metadata. Harnad and Carr [92] argued for open citation linking (OpCit project) to integrate, navigate and analyze e-print archives. Elmqvist [64] implemented CiteWiz a visualization tool for scientific citation networks.

[3] http://www.garfield.library.upenn.edu/histcomp/sciento_all_citing/index-aus-4.html

It has been recognized that the number of publications is not a good measure of scientific output. Instead, number of citations has been used as a more objective measure. Still this number may be inflated by self citations - when one of the co-authors is a co-author of the cited work. Banning self-citations is not constructive as these serve their purpose by pointing to related previous work. Instead self-citations are generally filtered out when citation indices are compiled. Yet number of citations is not comparable across disciplines and many break-through papers have been ignored at their time and became famous much later. Also some citations are negative – where another researcher publishes a criticism or a correction. Despite these disadvantages citation is the most commonly used piece of evidence in quality evaluation. A number of measures for authors and venues has been proposed. A description of the process by which highly cited authors are identified in Science Citation Index which involves manual name disambiguation is described in [51]. Garfield proposed the widely used, yet controversial, Impact factor [72] of venues, and Seglen was among his first critics [198] campaigning against its use. Other ranking schemes inspired by work in computer science are being used recently – Pagerank [28, 165] and hubs and authorities [54] have been deployed in citation databases to rank results of queries. Sidiropoulos and Manolopoulos [201] demonstrated and automated system for journal and conference ranking by impact using DBLP data.

Many ranking tables have been produced using various methodologies. Assessment of research quality on UK universities is used to distribute funding [98]. University ratings [82] help prospective students make a choice based partially on the scientific output of university staff. Oswald argued that the quality of university science teaching in the UK is being held back by a lack of world-class scientists working in UK institutions [164]. University of Illinois is one place that hosts a compilation of various rankings [159].

Newman investigated the structure of scientific collaboration networks. He considered two scientists to be connected if they have authored a paper together, and construct explicit networks of such connections using data drawn from a number of databases, including MEDLINE (biomedical research), the Los Alamos e-Print Archive (physics), and NCSTRL (computer science). We show that these collaboration networks form *small worlds* in which randomly chosen pairs of scientists are typically separated by only a short path of intermediate acquaintances. He further reported the mean and distribution of numbers of collaborators of authors, demonstrated the presence of clustering in the networks, and highlighted a number of apparent differences in the patterns of collaboration between the fields studied. [153]

The citation links can also be used to determine similarity of two documents. Kessler proposed bibliographic coupling [114] as a measure of similar-

ity. Kessler's bibliographic coupling was a binary relation containing all pairs of papers where both papers cite, at least one, same paper. More recently, bibliographic coupling is used in the sense of similarity which is normalized and computed as the fraction of the overlap of their bibliographies to the total size of the bibliographies. Cocitation defines a dual similarity to bibliographic coupling. Cocitation similarity, of two papers u and v, is greater the more papers cite both u and v [227]. Two documents are said to be co-cited if they appear simultaneously in the reference list of a third document. Lu et al. performed an extensive review of various link based similarity measures and compared them to similarity judged by human experts. The experiments were performed using CiteSeer citation graph. [139]

Lawrence [131] argued that papers available online are more cited than papers published on paper only. McGovern [145] studied high energy physics archive (HEP) and observed that authors prefer certain journals and topics.

The effect of geographical location on collaboration and scientific performance has been studied repeatedly. Batty [17, 16] manually identified the top 250 highly cited authors in ISI citation index. His study focused on medicine where he found high geographical clustering of authors and dominance of USA. Matthiessen and Schwarz [142] studied the geographical distribution of scientific centers in Europe. May [143] analyzed in his Science paper the Science citation index and compared the correlation of scientific performance, GDP and other normalized relative metrics. Kim [117] analyzed the Korean national specific citation patterns. They observed that the price of journal subscription affects its citation rates.

The effect of foreign language and foreign names on the reliability of manually or automatically compiled citation indices has been noted. Price [181] argued that papers by non-english speaking scientists are undercited. Kotiaho's [123] experiments demonstrated that English speakers tend to make more mistakes when recalling non-english names which can increase miscitations [122]. Simkin and Roychowdhury [202] observed that the frequency of mis-citations follows a power-law. They then proposed a model of accidental mis-citations and their spread through copying the bibliographic entries. They concluded that mis-citations are propagated by researchers who do not read the original paper.

Citation distributions

McGovern [145] also reported most cited authors in HEP and that 80% citations received by 26% authors. With regard to the distribution of citations,

Laherrere *et al* [126] argued that a stretched exponential[4] is suitable for modeling citation distributions as it is based on multiplicative processes and does not imply an unlimited number of authors. Redner [184] then analyzed the ISI and Physical Review databases and showed that the number of citations follows roughly two power-laws - one for highly cited papers and a different one for low cited papers. Lehmann [134] attempted to fit both a power law and stretched exponential to the citation distribution of 281,717 papers in the SPIRES [206] database and showed it is impossible to discriminate between the two models with the data. Tsallis and Albuquerque [36] fit a nonextensive curve to the citation distribution in ISI and PRE datasets. They argue that contrary to Redner's conclusions the phenomenon behind the citation distributions may be the same for both low as well as highly cited papers. There is no consensus yet on the exact form of prevailing mechanism behind citation distribution. Due to the noise in data and biases introduced during data collection, discrimination between models is hard. Given this uncertainty we adopt Redner's model for our comparisons as it is slightly easier to fit.

A complementary statistic to citation distribution (in-degree) is the distribution of number of documents cited in one paper (out-degree). This has been studied by Vasquez [220] who reported statistics for journals in the period 1991-1999. The out-degree distribution is characterized by a maximum at intermediate out-degrees. There are strong fluctuations from journal to journal. There two classes of journals. These two classes are associated with the existence or not of a restriction in the maximum number of pages per paper. The shape of the out-degree distribution did not change appreciably from period to period, but the average out-degree was observed to increase logarithmically with the number of published papers. Vasquez modeled the distribution using a recursive search model.

Much of the research on citation distributions has come from the Physics community. Surprisingly little work has been done on computer science papers. The ISI dataset contains computer science papers but these were usually studied together with other disciplines despite the fact that their dynamics may differ. The only work we are aware of [153] is based on a small dataset (13000 papers) and was concerned with the distribution of the number of collaborators.

[4]Stretched exponential distribution has the form $exp(-(x/w)^c)$ where w and c are the distribution parameters

4.1.3 Networks in computer science

Computer networks were relatively orderly and simple. This changed with the advent of the Internet and then the World Wide Web (Web). Internet itself is not controlled and does not have externally enforced topology. Similar anarchy rules on the Web. Still both of these networks exhibit regularities that surprised many. Researchers have explored several aspects of the networks which differ in the interpretation of the link between two vertices i) there is no direct interpretation and researchers study structure and its statistical properties ii) link is interpreted as an endorsement ii) link is used as a signal of similarity or relatedness iv) link is a means of communication or navigation. We review each of these areas below.

Structure of networks

Even though the Internet or Web do not have any particular structure centrally enforced and are created in a fully distributed manner, some sort of order spontaneously emerges. Scientists have been studying the structure of these fascinating networks from different perspectives. These structural studies may be characterized as descriptive, modeling, and visualization studies.

Broder et al. describe the bow-tie like structure of the web observed in the Altavista crawl. They present the sizes of individual components and other network properties such as diameter, in and out degree distributions, reachability (different from us). [29] Authors study AOL log files and two crawls from infuses and Internet Archive. They observe power-laws in number of inlinks, outlook's, size, and number of visitors. [6] They showed a power law in the distribution of number of pages in websites. They used Alexa and Infoseek crawls. [105]

Mathematical and generative models have been employed to predict the properties of the growing networks and to understand the mechanisms behind their growth.

The structure of the Web and Internet has been originally modeled as a random graph [65]. This model failed to explained the power-laws observed in degree distribution for example. Albert and Barabasi proposed their preferential attachment model (BA) [8] based on observations from a large scale crawl. Their model resulted in a power-law degree distribution. Through simulations they showed that the diameter of networks generated using this model grows with the logarithm of number of nodes. They extrapolated using parameters learned on their crawl, that the diameter of the whole Web is just 19. Suggesting the Web is more connected than assumed at the time. Further improvements have been proposed resulting in

more accurate models for other types of networks. Shi and Mondragon![230] proposed Positive-Feedback Preference (PFP), a non-linear preferential attachment model, which is so far the most accurate model for the AS-level topology of the Internet.

Visualization of the large and complex networks is a complicated task. There are many ways of projecting the network graph on paper which result in very different representations and stress different properties of the network. The vast number of vertices and edges poses computational and representational challenges. Cooperative Association for Internet Data Analysis (CAIDA[5]) provides interesting visualizations and tools for making these visualizations. A social network analysis tool, Pajek [15], also provides several visualization algorithms.

Link as endorsement

The interpretation of links as endorsement has been very successful. This notion has been originally used in citation analysis and later adopted in computer science.

Virtually all link based ranking algorithms in information retrieval are based on this interpretation. The Altavista search engine already provided results sorted by number of inlinks. Kleinberg [120] later described his algorithm for identifying authoritative pages (*authorities*) and good starting points for surfing (*hubs*). A good hub is a page that points to many good authorities and an authority is a page that is pointed to from many good hubs. Each page has two scores associated – hub score and authority score. These scores, initially uniformly assigned are iteratively updated until they converge sufficiently. The pages with highest authority and hubness scores are *authorities* and *hubs* respectively. PageRank [28, 165] is based on a similar idea, but does not divide pages into two groups. Instead a good page is the one that is pointed to by other good pages. The intuitive model behind this algorithm is a random surfer model which assumes that the surfer follows a random link from the page he is currently viewing or jumps to another random page. Ingversen defined web impact factor (Web-IF) similar to journal impact factors from the ISI citation index. However this computation relies on noisy data provided by search engines and has to be interpreted carefully. [108]

Thellwall [210] argued that a single page is not the element that should be studied, instead a web document that consists of related pages. Bharat et al. [18] studied the macro structure of the Web based on links between

[5] http://www.caida.org/

hosts and domains as opposed to individual pages and described a related host finding algorithm operating on this host graph.

The use of hyperlink structure for ranking search engine results motivated efforts known as link spam. Link spam is a way of artificially increasing ones pagerank by creating artificial hyperlink structures. In response to this prevalent problem, search engines currently employ variations of the TrustRank algorithm [86]. TrustRank uses a seed set of trusted pages which are reviewed by human expert. Other pages are then assigned TrustRank score which is diminishing with growing distance from the seed set.

Interpreting link as an endorsement brought many interesting results in the analysis of the Web. In this thesis we use this interpretation in the citation analysis in Chapter 5. In our analysis of websites we use the interpretation of a link as a means of navigation/communication which is more appropriate in that case – the links are internal links inside a website and serve mainly for navigation.

Link as similarity

In some situations an edge between two vertices has been created because these two vertices are similar – they may represent two friends who like the same sport or two pages on the same topic. This property is often algorithmically exploited.

Links have been used for clustering in networked environments. Chakrabarti et al. [43, 44] presented their automatic resource compiler (ARC) as a fully automatic system for generating lists of topically related resources. The output was similar to portals like Yahoo! Authors asked real users to judge the accuracy and coverage of the lists compiled by ARC and lists from Yahoo! and Infoseek. ARC did well in comparison and in some cases was even better than the manual lists. Gibson et al. [76] developed a notion of hyperlinked communities on the Web through an analysis of the link topology. By invoking a simple, mathematically clean method for defining and exposing the structure of these communities, they observed: The communities can be viewed as containing a core of central, authoritative pages linked together by hub pages; and they exhibit a natural type of hierarchical topic generalization that can be inferred directly from the pattern of linkage. Our investigation shows that although the process by which users of the Web create pages and links is very difficult to understand at a local level, it results in a much greater degree of orderly high-level structures. Kumar et al. [125] introduced an incremental technique for identifying complete bipartite subgraphs. Such graphs often correspond to communities or their cores. They discovered many unexpected yet sensible communities. Flake et al. [68] described a clustering

algorithm based on flows in networks [69]. Community in their definition is a group of nodes that have more links within the community than outside. They demonstrate the algorithm on web pages clustering application. This algorithm identifies minimum cuts in an augmented graph of the link structure which separates well connected components. He et al. [95] experimented with normalized-cut clustering for search engine results. Andersen and Lang [9] presented another algorithm for finding communities from seed sets using just local neighborhood and provided theoretical guarantees on the quality of the communities. Spectral methods [155] have also been used for clustering web pages.

Even the largest search engines do not manage to cover the whole indexable Web. Techniques allowing quick location and crawling of topically related information are crucial for both the global and niche search engines. Chakrabarti et al. [42] presented a focused crawler which crawls interconnected communities of similar content. Diligenti et al. [60] described a focused crawler that builds a model of content distribution to avoid pursuing short term goals at the expense of more promising but less obvious directions. URL ordering strategies have been studied by Cho et al. [47] who demonstrated the performance gain.

We use an interpretation of a link as similarity when we use social network to provide item recommendations in Chapter 7. Although links are routinely used as a signal of similarity, they haven't been fully exploited for item recommendation yet.

Link as communication/navigation

For the Web, Adamic et al. [106] propose a model of user surfing behavior. In this model a user keeps surfing to next page as long as the value of the current page exceeds a certain threshold. The model yields a probability distribution of the number of pages visited within a website. Adamic et al. verified the model by comparing it to empirical data. Their model offers an explanation of the power-law distribution in page hits observed. Caldas [37] argued that online academic network links correspond to offline collaboration networks.

Park et al. [168] simulated the effect of random faults and malicious attacks on the Internet topology. They observed that the Internet is highly resilient to random faults but it is rather susceptible to targeted attacks on the highly connected nodes. The average diameter of the Internet has been stable or even decreasing as the number of nodes has been increasing. The Internet has been becoming more robust to random failures over time, but has also become more vulnerable to attacks. For peer to peer networks [163]

their structure greatly affected their success or failure. Peer to peer networks have to balance efficient search and delivery of content, attack and censorship resilience, bandwidth management and other issues which are all directly related to the network topology. Links with limited capacity have been modeled using flows [69].

4.1.4 Conclusion

Studies of computer science networks have benefited greatly from the methods developed for social network analysis (see for example [157]). Many techniques inspired by social network analysis or developed independently are in use in computer science. There is a big overlap although the different vocabulary in the two areas hampers the exchange of ideas. Park [167] is one of the early attempts to bring these together – it presented a review of previous work on hyperlink analysis in computer science for social scientists.

4.2 Previous recommender systems work

Resnick and Varian coined the term recommender system in [186]. Recommender systems have been widely used to overcome information overload and to provide better user experience by personalizing recommendations to each user. Institutions as varied as online shops [138], universities [75] and dating sites [30] use recommenders. Popular examples of such systems are Amazon[6] [138], Yahoo! Movies[7], Movielens [8], Netflix[9], Tapestry [79], Jester [80], Ringo [215] and Yahoo! Music[10] that recommend books, movies, news articles, jokes and music. Montaner et al. [150] provided a taxonomy of recommender systems on the Internet, and Schafer et al. [195, 194] contain an overview of e-commerce recommendation applications.

Datasets

Crucial for the development of recommender systems is the availability of large datasets. There are several of them freely available. Jester [1] has a dataset of 100 jokes with ratings by 73,421 users. Movielens [2] is the de facto standard for collaborative filtering benchmarks. It contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 Movielens users

[6] http://amazon.com
[7] http://movies.yahoo.com
[8] http://movielens.umn.edu
[9] http://www.netflix.com
[10] http://music.yahoo.com

who joined Movielens in 2000. Microsoft anonymous web data dataset [151] records which areas of Microsoft website each user visited during a one-week timeframe in February 1998. An online dating agency dataset is available [30] and contains a dump of explicit ratings of users' profiles from 2006. Many datasets are proprietary and carefully guarded as they present competitive advantage of respective owners and contain sensitive information about users.

Collaborative filtering algorithms

One of most successful approaches for building recommender systems is collaborative filtering. A pure collaborative filtering algorithm has been built by a simple memory-based approach [27] or more sophisticated model-based methods including k-nearest neighbors methods [190, 59], dimensionality reduction [191, 192, 185, 57, 101, 100], probabilistic methods [27, 172, 221], data mining [7, 58] and other machine learning methods [141].

The oldest collaborative filtering algorithm is the User-user algorithm [96]. This algorithm makes the assumption that users who agreed on their taste in the past will probably like similar items in the future. It finds the k most similar users to the target user and makes recommendations based on what these similar user liked. Currently the most often used algorithm is the item-item algorithm [190] which is in a way dual to the User-user algorithm. It looks at the items target user liked in the past and recommends items similar to these items. The similarity between two items is computed based on the number of times they have been bought by the same user. Deshpande and Karypis [59] compared several item-item algorithms with cosine similarity and probability based similarity, and experimented with two optional steps: similarity normalization and row normalization. They also evaluated higher order models that build similarities between larger itemsets than just single items and showed that this leads to higher quality recommendations. Another fast yet reasonably accurate algorithm called Slope [135] has been proposed by Lemire and Maclachlan. Recently Wang et al. [221] proposed a unified probabilistic fusion algorithm in which they merged user-user and item-item algorithms.

Dimensionality reduction methods attempt to approximate the presumably noisy data matrix by a smaller matrix. The benefits of this approach are smaller memory footprint, faster computation and a possibly better generalization. There are several factorization methods available for this task. Sarwar et al. [191] used the Singular Value Decomposition (SVD) to capture latent relationships between customers and products. They used the SVD to produce a low-dimensional representation of the original customer-product space and then computed neighborhood in the reduced space. The

SVD algorithm performed poorly on extremely sparse datasets but in some cases outperformed traditional algorithms on Movielens dataset. The decomposition may be computed offline and in that way speed-up the algorithm's online performance. Still it has to be updated periodically. Sarwar et al. proposed an improved incremental version of the algorithm in [192]. Goldberg et al. [80] implemented a fast collaborative filtering method for performing recommendations based on Principal Component Analysis (PCA). In the offline phase they reduced dimensionality and then used the matrix eigenvectors in the online phase to produce recommendations. The computation is independent of number of users and very fast. They implemented a constant time joke recommender system using this algorithm. In [185] Rennie and Srebro presented a fast algorithm for maximum margin matrix factorization (MMMF). The idea of margin maximization is the same as the one behind Support Vector Machines [219]. They show that MMMF is better than any of the eight algorithms surveyed by Marlin [141]. The good performance of MMMF is attributed to the fact that ratings are not missing at random which is the situation where margin maximization is useful. DeCoste [57] speeded up the implementation of MMMF 15-20 times as compared to [185] and used ensembles of partially trained MMMF models to further reduce the training time and achieve higher quality predictions.

There are different probabilistic approaches to collaborative filtering. Shani et al. [199] modeled a recommender system as a Markov decision process. They propose a method for online update and management of the large state-space. Their system has been deployed at an online bookstore[11]. Pennock et al. [172] introduced a method called Personality diagnosis. They assumed that users belong to a certain personality class and enter their ratings with Gaussian error distribution. Cheng [46] implemented a recommender for Berkeley Digital Library [12] using a Bayes network.

Data mining approaches to collaborative filtering aim to infer rules from the historical data that could be used for recommendations. Agrawal and Srikant [7] presented a fast algorithm for mining association rules from commercial transaction databases. Their algorithm finds all rules with a specified minimum support and minimum confidence. This problem is related to the problem of similarity computation between itemsets in collaborative filtering. Their algorithm does not address the step of recommendation generation using such rules. Demiriz [58] presented an algorithm called e-VZpro based on mining associations from historical data and shows an improvement over other association mining algorithms and dependency networks. However,

[11] http://www.mitos.co.il
[12] http://elib.cs.berkely.edu/docs

item-item algorithm performed even better. The evaluation has been done on MSWEB and two internal Verizon customer datasets.

Many comparisons of collaborative filtering algorithms in terms o prediction precision have been performed. Each proposal of a new algorithm generally contains a comparison with the previous algorithms but there have been as well several papers focused on independent benchmarks methodology. Herlocker [97] surveys performance metrics used in collaborative filtering research evaluation and possible experiment designs. Breese et al. [27] compared several algorithms including techniques based on correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods. They compared the predictive accuracy of the various methods on real world datasets (EachMovie, MS web, Nielsen TV network data). Brozovsky and Petricek [30] compared user-user algorithm, item-item algorithm and baseline global popularity. They reported that on the real dataset from an online dating agency the user-user algorithm performed best. Cosley et al. [54] described a testbed for development of recommender systems for CiteSeer digital library. A daemon provided events generated by users of CiteSeer and solicited recommendations for the active user. Pennock et al. used metrics based on click-throughs and downloads of documents to evaluate their prototype recommenders. Sarwar et al. [193] performed offline benchmark of recommenders (association rules, SVD, different types of neighborhoods) on Movielens dataset and Fingerhut Inc. transaction history. Karypis [112] compares item-item algorithm variants with user-user algorithm on five datasets (e-commerce, order catalog, credit card transactions, personal skills from CVs, movie ratings). He tests several possibilities how to tune these algorithms and reports improvement of 27% in recommendation quality and 28% speed up as compared to traditional user-user. Sarwar et al. [190] compared several variants of item-item algorithms in terms of prediction error. They experimented with different similarity metrics for item-item similarity computation (cosine, adjusted cosine, correlation) and also with different ways of obtaining the recommendations (weighted sum, regression model). Regression performed worse than weighted sum and adjusted cosine performed better than pure cosine and correlation. They compared these variants to user-user and argued that item-item is superior to user-user variants in terms of performance and recommendation quality. Marlin et al. [141] provided a machine learning perspective on collaborative filtering. He proposed several new machine learning based algorithms and performed an evaluation of total eight algorithms on the Movielens dataset. Currently MMMF holds the record on the Movielens dataset in terms of prediction precision. Still simple item-item algorithm remains the most popular as it is easier to implement and train and provides good quality recommendations.

Scalability

The size of the rating matrix grows significantly with new users and items. In some situations mostly just new users arrive and number of items remains stable (Jester joke recommender of a static set of jokes), in others there is a relatively stable number of users who contribute vast amount of content (photo sharing community) but often there are both many new users and many new items. Even if the matrix remains sparse the amount of data presents a challenge to recommenders. Scalability has been a focus of the recommender research and several ways of scaling algorithms has been explored – i) matrix decomposition ii) subsampling and iii) clustering.

The matrix factorization methods [191, 185] discussed earlier achieve reduced memory requirements and increased speed by approximating the data matrix by a smaller matrix. The different factorization methods differ in their assumptions about missing data and MMMF has been shown to perform best in terms of accuracy [185]. Additional tricks such as bagging several sub-optimal instances have been used to further speed up the algorithm. Matrix factorization methods may generalize better if projections into a lower dimensional space select important dimensions and reduce noise.

Subsampling can reduce the size of the dataset and reduce noise if good representants are selected. Yu et al. [229] analyzed four sub-sampling methods for speeding up collaborative filtering and reported results of experiments on EachMovie dataset. In traditional item-item algorithm [59] the model size is controlled by the parameter k which effectively prunes the similarity matrix to keep only k most similar items for each item.

Svensson et al. [208] suggested manual labeling of user groups by an editor to help recommendations, while Ungar and Foster [213] described automated optimal user clustering for user-item matrix reduction. Ungar and Foster tested their system on synthetic data and real purchase data from CDNow and iReactor. Connor and Herlocker [148] proposed algorithm ROCK that performed item clustering and then made predictions within these clusters. They reported improved speed but mixed results on prediction quality. Kalgren [111] also implemented a news group filtering system based on user clustering.

Cold-start problem

A pure collaborative filtering algorithm often suffers from the cold-start problem when the rating matrix is sparse. This happens especially at the start of the deployment of a recommender system (*new-system cold-start*) and then again each time a new user joins the recommender (*new-user cold-start*).

Several approaches, including sub-sampling methods [229], clustering methods [213, 148] and matrix decomposition methods [191, 192, 185, 57], address scalability of collaborative filtering.

Wang et al. [221] presented a probabilistic algorithm that combines the idea of user-user algorithm with that of item-item. Makes prediction based on i) items similar to items the target user liked, ii) items that users similar to target user liked and iii) items similar to items of target user that were liked by users similar to the target user. Papagelis [166] used transitive similarity/trust to improve prediction quality of collaborative filtering in the cold start situation. Weng et al. [225] used recursive trust propagation model as a transitive similarity to provide recommendations which helped with cold start and improved quality of recommendations. Soliciting preferences is obtrusive and the users are willing to make only limited effort during the rating process. It is important to judge well for which items to solicit ratings as this may influence the performance of the recommender. Boutilier et al. [25] introduce a method for determining items for which to solicit ratings based on maximizing the EVOI (Expected Value Of Information). They propose an offline precomputation that will speed up the expensive online EVOI computation. Some systems try to avoid cold start problem and the problem of overall data sparsity by use of implicit preferences such as downloads, clicks or time spent reading a piece of news. Even though there are ways to partially reduce the cold-start problem, it remains an issue. For this reason, hybrid methods have been proposed, which combine collaborative filtering with some other way of making recommendations. We discuss these next.

Hybrid algorithms

Hybrid algorithms combine some other source of information with collaborative filtering. This work is related to our use of social network for item recommendations. In the past the following algorithms have been published.

Good et al [81] was the first to explore the idea of filterbots. Park et al. [169] implemented several filter-bots and demonstrated how they improve item recommendations even if the filter-bots are very simple – rating for example by genre only. Popescul et al. [179] exploited item content to improve the recommender performance and overcome the sparse dataset problem using a three-way aspect model of users, items and content. Burke et al. [33] implemented a content based hybrid collaborative filter for an online newspaper. They used knowledge-based technique to bootstrap the system while its data pool was small and they used the collaborative filter as a postfilter for the knowledge-based recommender. Burke and Robin [34] surveyed the land-

scape of existing and other possible hybrid recommenders, and introduced a new hybrid system, EntreeC, that combines knowledge-based recommendation and collaborative filtering to recommend restaurants. They show that semantic ratings obtained from the knowledge-based part of the system enhance the effectiveness of collaborative filtering. Claypool et al. [49] also implemented a different hybrid content based collaborative filtering approach for an online newspaper. Burke et al. used their knowledge-based algorithm to provide recommendations and item space navigation in [35, 32, 33]. Brunato and Battiti [31] implemented a location aware recommender system deployed in Trento, Italy. Pazzani [170] described a unified framework for hybrid algorithms combining item content, user demographic information and collaborative filtering. Middleton [147] described a web-based research paper recommender system using an ontology containing information automatically extracted from departmental databases available on the web. The ontology, containing information about users such as research interests, was used to address the recommender systems cold-start problem. Torres et al. [211] evaluated their hybrid algorithms through offline experiments on a database of 102,000 research papers, and through an online experiment with 110 users. For both experiments they used a dataset created from the CiteSeer repository of computer science research papers and they recruited American and Brazilian users to test for cross-cultural effects. They observed that different algorithms are more suitable for recommending different kinds of papers, and that users with different levels of experience perceive recommendations differently.

User related issues of recommender systems

In addition to the algorithmic issues there are several user related issues concerning i) privacy, ii) system robustness to attacks, iii) user perceptions of recommendations and iv) human computer interaction (interfaces).

By revealing their preferences users give up some of their privacy. This is an issue especially for items that have few ratings as the users are easier to identify. Unfortunately, these are the items for which the recommenders need additional ratings the most. Canny [39] proposed a collaborative filtering algorithm called Sparse Factor Analysis that protects user privacy and compares favorably to SVD. The same author explored the use of Homomorphic Encryption to protect privacy of users [38].

The behavior of users and their buying decisions are significantly influenced by the recommendations which they receive. This provides incentives for unscrupulous marketers to *shill* recommenders by providing false preferences and reviews. O'Mahony et al. [160, 161] pointed out the risk

of attacks to recommender systems by inserting malicious users. Lam and Riedl [127, 128] describe several ways to attack a recommender system using fake users who rate items in such a way that they appear similar to as many users as possible and then manipulate the rating of the target item. They present a classification of attacks into *nuke*, and *push* attacks. Nuke attacks aim to decrease rating of a competing product while push attacks aim to boost the rating of item which the attacker wants to promote. Lam and Riedl conclude that administrators of recommenders have to watch out but that traditional metrics do not reveal much about the presence of attackers.

Offline benchmarks capture just a part of the reality. User studies are necessary to fully evaluate recommender systems. Several user studies have been performed in addition to previously discussed offline benchmarks. Kirsten et al. [183] performed a usability study of over 20 recommender systems. Rashmi had users try out different recommendation systems, from Amazon to MediaUnbound and gathered qualitative and quantitative data. The report doesn't describe the tests, the questions, how the data was analyzed, etc. Ziegler et al. [231] showed how topic diversification in the recommended set increases recommendation quality as perceived by users. Also, the reliability of recommendations, as perceived by users, can be improved by explaining the reasons for the recommendation [54, 203, 23]. This work is somewhat orthogonal to our own. Brozovsky and Petricek [30] performed a user study with 111 participants who compared anonymous recommendations of user profiles in the setting of online dating agency. Recommendations were provided by collaborative filtering algorithms and baseline popularity algorithm and random profiles. Collaborative filtering outperformed all baseline algorithms. Geyer-Schulz and Hahsler [74] performed a study in which they compared recommendations generated by association rules to recommendations by human experts. They have shown that the best algorithms reach 70% accuracy and 60-90% precision if human experts are taken as a ground truth. Hayes et al. [94] argued that user satisfaction with a recommendation strategy can only be measured in an on-line context. They proposed an evaluation framework which involves a paired test of two recommender systems which simultaneously compete to give the best recommendations to the same user at the same time. The user interface and the interaction model for each system is the same. Sinha and Swearingen [203] evaluated the quality of recommendations and usability of six online recommender systems – three book recommender systems (Amazon.com, RatingZone and Sleeper) and three movie recommender systems (Amazon.com, MovieCritic, Reel.com). Quality of recommendations was explored by comparing recommendations made by recommender systems to recommendations made by the users friends. Results showed that the users friends consistently provided bet-

ter recommendations than RS. However, users did find items recommended by online recommender systems useful: recommended items were often new and unexpected, while the items recommended by friends mostly served as reminders of previously identified interests. Bonhard et al. [23] analyzed how does familiarity, profile similarity and rating overlap affect users' trust. Bonhard and Sasse [24] executed a qualitative study with 44 participants the relationship between the advice seeker and recommender proved important. They argued that recommender systems must establish a connection between the advice seeker and recommenders through explanation interfaces.

Interface design greatly influences user experience as well as the performance of recommender systems. Ginty and Smyth [78] advocated the use of comparison-based feedback solicitation in online recommender systems and iterative item recommendation. They compared their approach to casual conversation and argue that it is less expensive for the user to provide feedback this way. Cosley et al. [53] studied two aspects of recommender system interfaces that may affect users opinions: the rating scale and the display of predictions at the time users rate items. They found that users rate fairly consistently across rating scales. Users can be manipulated, though, tending to rate toward the prediction the system shows, whether the prediction is accurate or not. However, users can detect systems that manipulate predictions. Chen and Singh [45] computed reputation from ratings to help users evaluate the reliability of recommendations. Guha [85] presented a computational model of trust and showed how it can be used to identify high quality content in an Open Rating System, i.e., a system in which any user can rate content. He presented a case study – Epinions.com – of a system based on this model and describe a new platform called PeopleNet for harnessing this phenomenon in an open distributed fashion.

4.3 Network analysis and recommender systems

Methods of social network analysis have been applied to recommender systems [203, 113]. In these studies, the social network often serves as a means of establishing reliability of recommendations or as a defense against attacks in distributed systems. Referral Web [113] was the first system that combined social networks and collaborative filtering. It used a social network, mined from co-occurrences of names on web pages, to rank matching experts by their distance from the active user. The main purpose was to help users find experts on a topic in their social neighborhood. McDonald [144] imple-

mented recommender of colleagues with expertise for potential collaboration – he concluded that social networks in groupware applications often do not match the social network as perceived by users. Weng *et al* [224] and Papagelis et al [166] constructed an artificial social network based on user-user similarity. They employed a recursive trust propagation model to compute trust between users that have no common items rated. The dense trust matrix then served to provide predictions using pairs of users that do not have any co-rated items. This alleviated the cold start problem and improved prediction accuracy as measured by the mean absolute error. Rashid et al. [182] proposed an algorithm-independent definition of influence that can be applied to any ratings-based recommender system. They showed experimentally that influence is expensive to compute exactly but may be effectively estimated using simple, inexpensive metrics. On the border with social network analysis – Adamic et al. [5] performed an analysis of the Nexus social network and observed varying homophily. Distributed recommender system where different recommenders act as agents. Uses trust. [118] Authors advocate the need of trust in a distributed recommender system. [149]

Early collaborative filtering systems, such as Ringo music recommender [200], made references to social collaborative filtering. This did not necessarily mean that the algorithms took explicitly advantage of the social network. Generally this means that the whole community collaborated by sharing ratings. This differs from our notion of social recommender which uses real social network to provide recommendations.

4.4 Summary

To recapitulate – contributions of this thesis are in 1. website structure analysis, 2. citation analysis and 3. use of social network for item recommendations.

1. Our website structure analysis is novel in that it focuses on previously neglected class of websites - the e-government websites, foreign offices in particular. There have been very few user based studies where users could be observed in a controlled environment.

2. Our citation analysis compares for the first time two citation databases with focus on the different acquisition methods and the resulting biases. It is also one of rare computer science centered studies so far.

3. The use of social network for item recommendations and especially the combination of social network information with traditional collab-

orative filtering recommenders is following a previously unexplored direction. This research has been made possible thanks to the recent availability of large datasets of network and preference data. Unlike hybrid previous recommender systems this approach does not use item content or user demographic data. Instead a qualitatively new type of data associated with users is exploited – real social network.

Chapter 5

Citation analysis

In this chapter we investigate citation networks and attribute data in two computer science citation databases. We compare the citations networks to those analyzed in Physics community. We show that even networks in similar context exhibit different network properties. We also describe and model a type of bias introduced by different paper acquisition methods. Results in this chapter have been published in [175] and [176].

Several public[1] databases of research papers became available due to the advent of the Web [10, 132, 56, 52, 50, 55, 197]. These databases collect papers in different scientific disciplines, index them and annotate them with additional metadata. As such, they provide an important resource for (i) finding publications, (ii) identifying important, i.e. highly cited, papers, and (iii) locating related papers. In addition, author and document citation rates are increasingly being used to quantify the scientific impact of scientists, publications, journals and funding agencies. Grant applications as well as job applicants are both evaluated based on their publication record as compared to that of competitors. Distribution of resources is often driven by publication success measured using databases such as those studied in this chapter.

Within the computer science community, there are two popular public citation databases. These are DBLP [56] and CiteSeer [132]. The two databases are constructed in very different ways. In DBLP, each entry is manually inserted by a group of volunteers and occasionally hired students, who manually obtain entries from conference proceeding and journals. In contrast, each entry in CiteSeer is automatically entered from an analysis of documents found on the Web. There are advantages and disadvantages to both methods and to our knowledge the effect of the different acquisition

[1] By public, we mean that access to the database is free of charge. Commercial databases are also available, the most well-known being the science-citation index [196]

methods has not been analyzed before.

In this chapter we analyze and compare the data from CiteSeer and DBLP citation databases. We study both the attribute data such as publication year and number of authors as well as the citation network data. We show significant differences between the two Computer Science citation databases. We introduce three simple probabilistic models, that aim to explain the bias introduced by the fundamental differences in acquisition methods between the two databases. We also replicate some of the previous citation distribution studies and show that citation distributions from both DBLP and CiteSeer differ considerably from those reported other research communities. It is important to understand the differences and limitations of citation databases as they are more and more used to make critical decisions regarding the direction of future research.

This chapter is organized into four parts. First we review the related work that used similar datasets in Section 5.1. In Section 5.2 we describe the DBLP and CiteSeer datasets. We then compare the two databases in Section 5.3. We analyze the differences introduced by different acquisition methods (Subsection 5.3.1) and we reveal that there are very pronounced differences in the citation distribution (Subsection 5.3.2). Finally we summarize results of this chapter in 5.4.

5.1 Previous citation DBLP and CiteSeer work

The number of citations is the most widely used measure of academic performance and as such it influences decisions about distribution of financial subsidies. The study of citation distributions helps us understand the mechanics behind citations and objectively compare scientific performance. There has been considerable work in the area of citation analysis and a review of this work has been presented in section 4.1.2. Here we review in more detail previous studies of CiteSeer and DBLP - the two dataset that we use.

Giles et al. introduced CiteSeer in [77] as their autonomous citation indexing system which indexes academic literature in electronic format (e.g. PDF and Postscript files on the Web). CiteSeer understands how to parse citations, identify citations to the same paper in different formats, and identify the context of citations in the body of articles. More details on the inner workings of CiteSeer have been revealed in [132] and the free availability of the software behind the library was announced in [207]. Bollacker et al. [22] posed CiteSeer system in the framework of autonomous agents.

DBLP has been founded and operated by Michael Ley [137]. Ley and his coauthors presented history of DBLP and its inner workings in [140]. Ley

and Reuther compared the data quality of DBLP to that of CiteSeer and Google Scholar and discussed the massive human effort behind the effort to preserve this data quality [136].

Our work is not the first study of the CiteSeer or DBLP – it has been preceded by publications in information retrieval, machine learning and computer–human interaction as discussed below.

Researchers in the information retrieval community have experimented mainly with DBLP. Neves and Adrian [152] studied the problem of answering queries that require query splitting and queries that cover several related topics on DBLP corpus. In [71] Ganti et al. used DBLP for clustering experiments.

DBLP has been popular with XML research community as all the data from DBLP has been available in the form of compressed XML files. Lee et al. [133] used DBLP to study the functional XML dependencies. DBLP has been extensively used in XML benchmarks. An XPath engine described in [171] by Peng et al. was benchmarked on several datasets including DBLP XML data. Yao et al. [228] used DBLP in their benchmark of XML database management systems. Kim and Fox [116, 84, 115] have described a distributed hybrid search engine, SphereSearch, that searches both metadata as well as unstructured text. They also performed a benchmark of the system on datasets including the DBLP. Filtering system for XML streaming is benchmarked by Uchiyama et al. [212] on several datasets including DBLP XML data.

In citation databases there are several sources of possible ambiguity. There are some very common names that may be used by multiple authors leading to their citation records being merged. On the other hand authors may change their names, vary the use of diacritics, use pseudonyms, alternate the use of initials, etc. This results in publications of a single author being split into artificial groups. Similar ambiguities are introduced in virtually any metadata field in digital libraries by human error, OCR misclassification or software bugs. Both DBLP and CiteSeer have been frequently used in disambiguation experiments. DBLP was generally used as a ground truth due to its extremely low error rate, while CiteSeer has been usually used as data that need disambiguation. Han et al. [88] described and compared two supervised learning techniques for author name disambiguation i) a naive Bayes probabilistic model and ii) Support Vector Machines. As features Han et al. used author names, paper and venue titles from DBLP and CiteSeer. In [89] Han et al. presented an unsupervised learning approach using K-way spectral clustering that disambiguates authors in citations. Their approach utilized three types of citation attributes: co-author names, paper titles and publication venue titles. They measured the performance of the algorithm

on 16 different name datasets, author home pages and citations collected from the DBLP database. Bhattacharya and Getoor [19] described a general method of identifying records with the same information in the presence of noise. They used an iterative algorithm that takes into account the context of record tuples for similarity computation. Their algorithm showed better accuracy at the cost of longer computation time. B. W. On et al. [162] test a two-step method for author disambiguation on data from DBLP. In the first step author names are divided into blocks and then a pairwise disambiguation is performed inside each block only. Hong et al. [103] describe another system called OpenDBLP that performs disambiguation. They present a proof of concept implementation that uses the DBLP data. Tan et al. [209] proposed a name disambiguation method using hostnames returned by search engine when searching for a citation. They performed a k-means clustering based on the top 10 hostnames returned for each citation. Hassell et al. [93] used DBLP to construct an ontology for disambiguation of DBWorld message archives. On CiteSeer, Bhattacharya and Getoor [20] evaluated a generative unsupervised author disambiguation algorithm, Jiang et al. [110] used a clustering approach to citation disambiguation and implemented a wrapper clustering results from CiteSeer. Machine learning techniques for citation disambiguation have been employed by Lawrence et al. [129], who compared two distance metrics based on i) word and phrase matching and ii) string edits for citation disambiguation in CiteSeer.

There have been some studies of the link structure of the citation graph. Lawrence et al. [130] performed identification of hubs and authorities, similar documents, overlapping documents in CiteSeer and they also discuss error correction. Hopcroft et al. [104] describe an agglomerative link based clustering algorithm and applied it to the citation graph of CiteSeer database. They identified the stable communities under this clustering to correspond to natural communities. Authors investigated various link based similarity measures and compare them to similarity judged by two of the authors. The experiment was performed on CiteSeer citation graph [139].

Both CiteSeer and DBLP faced interface design challenges. Klink et al. [121] present DBLP interface for browsing publications by authors, coauthors, dates, and other metadata. CiteSeer interface is also constantly evolving. Cosley et al. [54] compared several recommender systems in a user study executed on a live CiteSeer system and some of the recommenders were included in the interface since. Bollacker et al. [21] implemented a system based on interest-profiles for tracking new publications that match the research interests of users. Recently Petinot et al. [173, 174] enabled CiteSeer as part of the semantic web through introduction of their application interface. CiteSeer-API is SOAP/WSDL based and allows for easy programatical

access to all the specific functionalities offered by CiteSeer services, including full text search of documents and citations and citation-based document discovery.

Two other studies fairly unrelated to the other studies have been published. Haase et al. [87] used DBLP data to simulate peer to peer semantic network and effectiveness of routing queries in such a network. Krottmaier proposed in [124] an automatic system using information from DBLP and CiteSeer to distill the current research interests of potential referees.

Surprisingly, there have been few studies of the citation networks as related to citation distributions. Some of the previously mentioned studies used the citation link structure but usually to perform clustering or similar tasks and ignored the network properties. The most closely related work to ours have been [131] and [201]. Lawrenece et al. [131] compared papers in CiteSeer and DBLP and argued that papers available online are more cited than papers published in printed form only. Sidiropoulos and Manolopoulos propose new ways to automatically rank journals and conferences by their impact, which they demonstrated on DBLP data [201]. Still, none of the studies that used both CiteSeer and DBLP databases focused on issues of the bias introduced by acquisition methods and the resulting differences between these two datasets. In addition, to our knowledge, there have been no study comparing citation distributions between CiteSeer, DBLP and other datasets. Before we move on to the analysis of CiteSeer and DBLP, we describe the two databases and the particular snapshots we used in more detail in the next section.

5.2 Datasets

There are a number of public, on-line computer science databases [10, 132, 56, 52, 50, 55]. The CS BiBTeX database [55] contains a collection of over 1.4 million references. However, only 19,000 entries currently contain cross-references to citing or cited publications. The Compuscience database [50] contains approximately 400,000 entries. The Computing Research Repository CoRR [52] contains papers from 36 areas of computer science and is now part of ArXiV [10] that covers Physics, Mathematics, Nonlinear Sciences, Computer Science and Quantitative Biology. Networked Computer Science Technical Reference Library is a repository of Computer Science Technical Reports located at Old Dominion University. The sizes of the databases together with their coverage are summarized in Table 5.1.

In addition to databases reviewed in Table 5.1, many publishers begin to make their content electronically available, although generally for a fee.

citation database	approximate size references (r) documents (d)	disciplines
CSBibTex	1,400,000 (r)	Computer Science
DBLP	550,000 (r)	DB, LP, IR, ...
CiteSeer	716,797 (d)	Computer Science
CompuScience	400,000 (r)	Computer Science
Arxiv	411,000 (d)	Physics, Math, CS, ...
CoRR	75,000 (d)	Part of Arxiv

Table 5.1: Size of several citation databases and scientific disciplines they cover. References (r) correspond to bibliographic records that may describe publications not available electronically or hidden in a proprietary database. Documents (d) refer to full texts of publications. Generally citation databases contain many more references than full documents.

We chose to examine DBLP and CiteSeer due to the availability of detailed citation information and their popularity. In our analysis we focus on the difference in data acquisition and the biases that this difference introduces.

5.2.1 DBLP

DBLP was created by Michael Ley in 1998 [56]. As of 2005 DBLP contained over 550,000 computer science references from around 368,000 authors. Papers in DBLP originally covered database systems and logic programming. Currently DBLP also includes theory of information, automata, complexity, bioinformatics and other areas. Database entries are obtained by a limited circle of volunteers who manually enter tables of contents of journals and conference proceedings. The volunteers also manually entered citation data as part of compiling the ACM anthology CD/DVDs. Corrections that are submitted to the maintainer are also manually checked before they are committed to the live database. Though the breadth of coverage may be more narrow than CiteSeer, DBLP tries to ensure comprehensive and complete coverage within its scope. The coverage of ACM, IEEE and LNCS is around 80– 90%. The narrower focus of DBLP is partially enforced by the cost associated with manual entry. Although there is the possibility of human error in the manual process of DBLP, its metadata is generally of higher quality than automatically extracted metadata[2].

[2] This remains true, despite the recent improvement of automatic extraction algorithms by use of support vector machines [90].

DBLP contains both types of data – i) attribute data such as publication year or paper authors and ii) network data in the form of citations between papers. We denote the DBLP dataset by $\mathbb{D}$ in the upper left index. The citation graph of DBLP is then denoted by ${}^{\mathbb{D}}\mathcal{G}({}^{\mathbb{D}}\mathcal{V}, {}^{\mathbb{D}}\mathcal{E})$. Attribute data is of different types which we distinguish by the upper right index. Capital Y in the upper right index means publication year data and upper right index of A represents authorship information. We therefore have the publication year data ${}^{\mathbb{D}}\mathcal{A}^{Y}({}^{\mathbb{D}}\mathcal{V}, {}^{\mathbb{D}}\mathcal{I}^{Y}, {}^{\mathbb{D}}\mathcal{F}^{Y})$ and the authorship data ${}^{\mathbb{D}}\mathcal{A}^{A}({}^{\mathbb{D}}\mathcal{V}, {}^{\mathbb{D}}\mathcal{I}^{A}, {}^{\mathbb{D}}\mathcal{F}^{A})$. ${}^{\mathbb{D}}\mathcal{F}^{A}$ denotes the matrix assigning to each paper in DBLP its authors. ${}^{\mathbb{D}}f^{A}_{v,a}$ is one if and only if a is author of paper v. The citation graph $\mathcal{G}^{D}(\mathcal{V}^{D}, {}^{\mathbb{D}}E)$ consists of scientific papers ${}^{\mathbb{D}}\mathcal{V}$ contained in DBLP connected by citations captured by matrix ${}^{\mathbb{D}}E$ where ${}^{\mathbb{D}}e_{uv}$ is one if and only if paper u cites paper v. Note that in the case of citations the relation $\mathcal{E}$ is purely asymmetric for scholarly articles as the papers do not cite themselves and also cannot cite each other mutually. If a paper cites other paper it has to be newer and therefore cannot be cited by the original paper.

Citation linking in DBLP was a one-time project performed as a part of the 'ACM SIGMOD Anthology' - a CD/DVD publication. The citations were entered manually by students paid by ACM SIGMOD. Because this was a one-off effort, DBLP now contains a significant number of new papers that have not been included in this effort. To mitigate against this distortion, we limit our citation analysis in both datasets to papers that have been cited at least once (CiteSeer 100,059 papers, DBLP: 10,340 papers).

In our analysis of the attribute data we used a DBLP dataset consisting of 496,125 entries. From this we extracted a dataset of 352,024 papers that specified the year of publication and the number of authors. Only papers published between 1990 and 2002 were included, due to the low number of papers available outside of this range. Table 5.2 provides a summary of the dataset size and comparison with CiteSeer dataset.

5.2.2 CiteSeer

CiteSeer was created by Steve Lawrence and C. Lee Giles in 1997 [132]. As of 2005 it contained over 716,797 documents. Automatic crawlers have the potential of achieving higher coverage as the cost of automatic indexing is lower than for manual entry. However, differences in typographic conventions make it hard to automatically extract metadata such as author names, date of publication, etc.

CiteSeer acquires entries in two ways. First, the publication may be encountered during a crawl. CiteSeer is not performing a brute force crawl

of the web but crawling a set of starting pages to the depth of 4-7[3]. In this case, the document will be parsed, and title, author and other information will be entered into the database. Second, during this parsing operation, a document's bibliography is also analyzed and previously unknown cited documents are also entered into the database.

CiteSeer is continuously updated with user submissions. As of 2005 updates were performed every two weeks. However, CiteSeer was not updated at all during the period from about March 2003 to April 2004. Prior to March 2003 crawls were made with declining regularity. As of July 2004 CiteSeer has been continuously crawling the web to find new content using user submissions, conference, and journal URLs as entry points.

In our analysis, we used a dump of CiteSeer dataset consisting of 575,068 entries. From this dump we extracted a dataset of 325,046 papers that specified the year of publication and the number of authors. Once again, similar to DBLP, only papers published between 1990 and 2002 were considered. It is also important to note that this dataset only contained entries that CiteSeer acquired by parsing the actual document downloaded from the Web, i.e. documents that were only cited but not actually parsed, were not included. We assume that CiteSeer parsing errors are independent of the number of authors and do not introduce any new bias.

CiteSeer contains data of both types – i) attribute data such as publication year, authorship, etc. and ii) network data consisting of citations between papers. We denote the CiteSeer dataset by capital blackboard bold letter $\mathbb{C}$ in the upper left index. The citation graph is then denoted ${}^{\mathbb{C}}\mathcal{G}({}^{\mathbb{C}}\mathcal{V}, {}^{\mathbb{C}}\mathcal{E})$. We distinguish attribute data of different types by an upper right index – the index Y for publication year data and A for authorship information. We therefore have ${}^{\mathbb{C}}\mathcal{A}^{Y}({}^{\mathbb{C}}\mathcal{V}, {}^{\mathbb{C}}\mathcal{I}^{Y}, {}^{\mathbb{C}}\mathcal{F}^{Y})$ the publication year data and ${}^{\mathbb{C}}\mathcal{A}^{A}({}^{\mathbb{C}}\mathcal{V}, {}^{\mathbb{C}}\mathcal{I}^{A}, {}^{\mathbb{C}}\mathcal{F}^{A})$ the authorship data. Table 5.2 provides a summary of the dataset size and a comparison with the DBLP dataset.

CiteSeer may be considered a form of self-selected on-line survey - authors may choose to upload the URL where their publications are available for subsequent crawling by CiteSeer. This self-selection introduces a bias in the CiteSeer database that we discuss later. A fully automatic scientometric system is also potentially susceptible to "shilling" attacks, i.e. authors trying to alter their citation ranking by, for example, submitting fake papers citing their work. This later issue is not discussed further in this thesis, but appears related to similar problems encountered by recommender systems [127].

[3]Personal communication with Isaac G. Councill, CiteSeer administrator.

Dataset	denoted by	documents	years
CiteSeer	$\mathbb{C}$	$\lvert^{\mathbb{C}}\mathcal{V}\rvert$ =325,046	1990–2002
DBLP	$\mathbb{D}$	$\lvert^{\mathbb{D}}\mathcal{V}\rvert$ =352,024	1990–2002

Table 5.2: CiteSeer and DBLP datasets used in this study. This data has been obtained from the respective dumps by processing described in Subsections 5.2.1 and 5.2.2.

5.3 CiteSeer and DBLP comparison

While both the DBLP and CiteSeer databases contain computer science bibliography and citation data, their acquisition methods greatly vary. In this section we first discuss these differences in acquisition methods. We first analyze the attribute data and then compare the citation networks present in the datasets. For attribute data we analyze the distribution of papers over time in each dataset, and the distribution of the number of authors per paper.

5.3.1 Attribute data analysis

We analyze the attribute data (metadata) available in the two datasets, particularly the publication year of papers $\mathcal{A}^Y$ and authorship information $\mathcal{A}^A$.

Accumulation of papers per year

We start off with the publication year data $\mathcal{A}^Y$. In order to compare the two databases, we first examined the number of publications in the two datasets for the years 1990 through 2002. These years were chosen to ensure that a sufficient number of papers per year is available in both datasets.

Figure 5.1 shows a considerable difference in the number of papers present in the two databases on an annual basis.

The increase in the papers per year exhibited by DBLP is explained by a combination of (i) the increasing number of publications each year [181, 140] and (ii) an increase in the coverage of DBLP thanks to additional funding and improvement in processing efficiency[4].

The decrease in the number of papers per year exhibited by CiteSeer since 1997 is mainly due to (i) declining maintenance; although (ii) declining coverage, (iii) intellectual property concerns, (iv) dark matter effect – when papers are hidden inside proprietary databases which are not normally

[4] Personal communication with Michael Ley

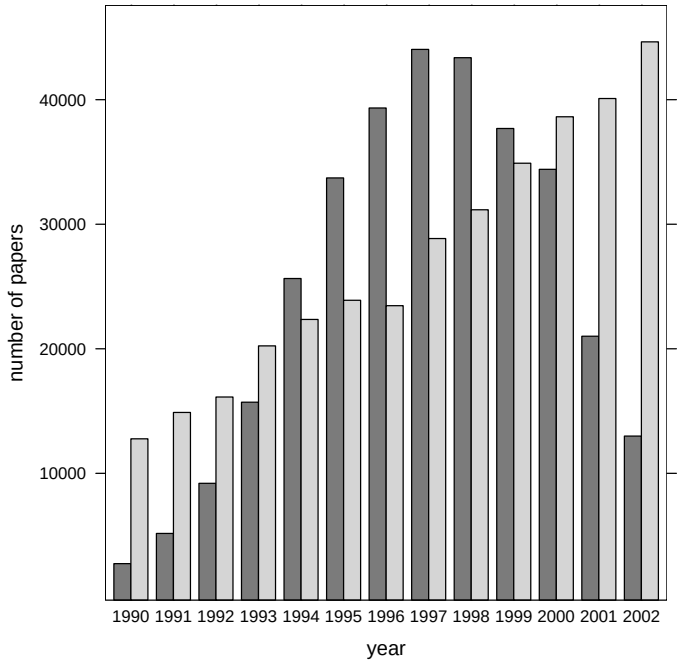


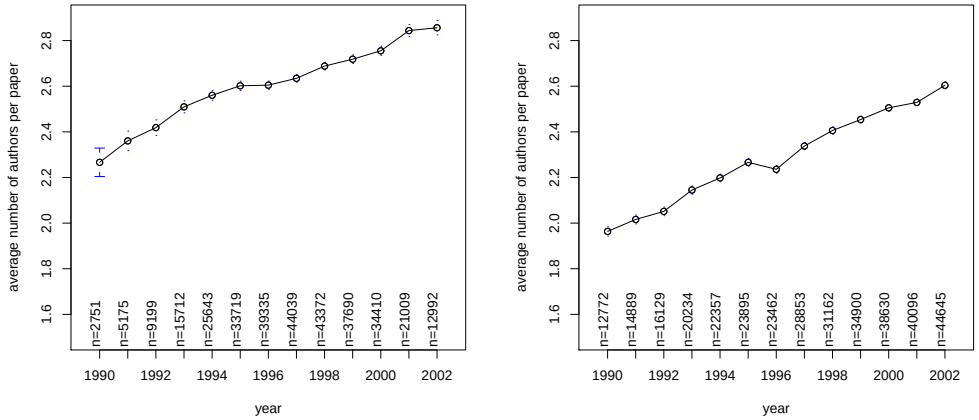
Figure 5.1: Number of papers published in the years from 1990 to 2002 present in the DBLP (light) and CiteSeer (dark) databases.

crawled and indexed [11] (v) end of web fever and (vi) specifics of submission process which may also have contributed. This comparison shows that although autonomous crawling has the potential to achieve higher coverage, it still requires regular maintenance.

Team size

We now proceed with the analysis of the authorship information $\mathcal{A}^A$.

We observe that the CiteSeer database contains a higher number of multi-author papers. We examined the average number of authors for papers published between 1990 and 2002, see Figure 5.2. In both datasets, the average is seen to be rising. We can see from the confidence intervals that each year observed the rise has been statistically significant with the only exception of 1996 when the average number of authors in DBLP dropped. This is probably due to inclusion of a large number of new venues with low average number of authors, which temporarily affected the overall mean. This rise in multi-authorship can be explained by (i) funding agencies preference to fund collaborative research and/or (ii) the fact that collaboration has become



(a) CiteSeer (b) DBLP

Figure 5.2: Average number of authors per paper for the years 1990 to 2002 in the CiteSeer and DBLP datasets.

easier with the increasing use of email and the Web.

Bias in number of authors

We look at the distribution of number of authors in more detail. Figure 5.3 examines the relative frequency of n_a-authored papers in the two datasets. Note that the data is on a log-log scale. We see that CiteSeer has far fewer single and two authored papers (y-axis is logarithmic and so the absolute difference is compressed in the plot). In fact, CiteSeer has relatively fewer papers published by one to three authors. This is emphasized in Figure 5.7 in which we plot the ratio of the frequency of n_a-authored papers in DBLP and CiteSeer for one to fifty authors. Here we see the frequency of single-authored papers in CiteSeer is only 77% of that ocurring in DBLP. As the number of authors increases, the ratio decreases since CiteSeer has a higher frequency of n_a-authored papers for $n_a > 3$. For high number of authors, especially $n_a > 30$, there is not enough papers available and the ratio is dominated by noise. We therefore limit our analysis to numbers of authors where there we have at least 100 papers in each dataset. This restricts the number of authors to less than 17.

There is an obvious cut-off from the power law for papers with low number of authors in Figure 5.3. For CiteSeer, we hypothesize that (i) papers with

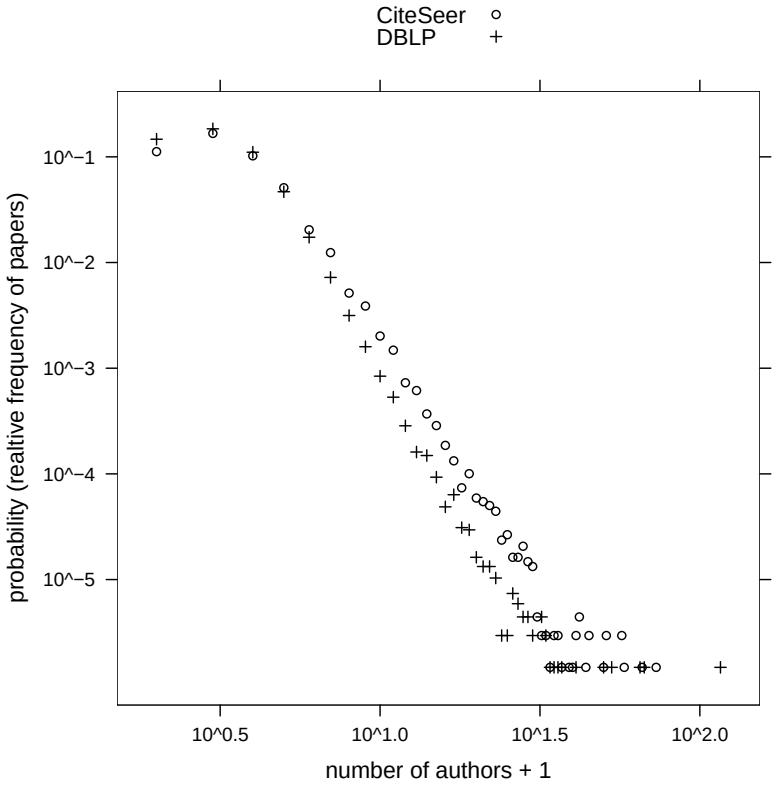


Figure 5.3: Probability histogram of number of authors (double logarithmic scale). The number of authors approximately follows a power law corresponding to a line with slope -0.23 for DBLP and -0.24 for CiteSeer.

more authors are more likely to be submitted to CiteSeer and (ii) papers with more authors appear on more homepages and are therefore more likely to be found by the crawler. These ideas are modeled in the next section. However none of these factors is relevant to DBLP, which also exhibits a similar drop off in single-authored papers. Other explanations may be that (i) single author papers are less likely to be finished and published, (ii) funding agencies encourage collaborative and therefore multi-authored research and (iii) it is an effect of finite size of the scientific community [126].

DBLP and CiteSeer data acquisition models

To explain the apparent bias of CiteSeer towards papers with larger numbers of authors, we develop two possible models for the acquisition of papers within CiteSeer. We also provide a simple acquisition model for DBLP. The

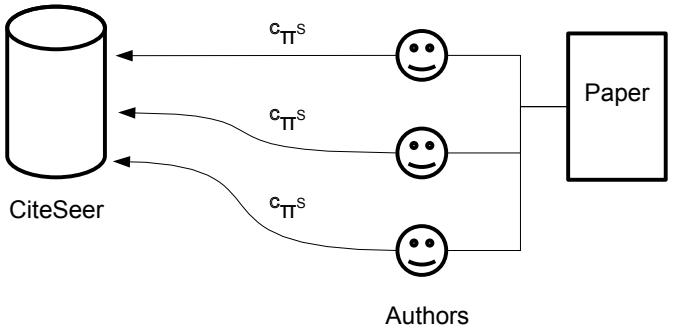


Figure 5.4: CiteSeer submission acquisition model. Each author submits a copy of the paper directly into CiteSeer with equal probability.

first CiteSeer model is based on authors submitting their papers directly to the database. The second CiteSeer model assumes that the papers are obtained by a crawl of the Web. We show that in fact, both models are equivalent.

To begin, let ${}^{\mathbb{C}}n(i)$ be the number of papers in CiteSeer dataset that have i authors, ${}^{\mathbb{D}}n(i)$ the number of papers in DBLP dataset that have i authors and let all(i) be the number of papers that have i authors published in all Computer Science. We remind the reader that $\mathbb{C}$ represents CiteSeer and $\mathbb{D}$ represents DBLP.

For DBLP, we assume a simple paper acquisition model illustrated in Figure 5.6. In the *DBLP model* we assume that there is a probability ${}^{\mathbb{D}}\pi$ that a paper is included in DBLP and that this probability is equal for all papers and therefore independent of the number of authors (Equation (5.1)).

For CiteSeer we assume that the acquisition method introduces a bias such that the probability, ${}^{\mathbb{C}}\pi(i)$ that a paper is included in CiteSeer is a function of the number of authors of that paper (Equation (5.2)).

$$ {}^{\mathbb{D}}n(i) = {}^{\mathbb{D}}\pi \cdot \text{all}(i) \tag{5.1} $$

$$ {}^{\mathbb{C}}n(i) = {}^{\mathbb{C}}\pi(i) \cdot \text{all}(i) = {}^{\mathbb{C}}\pi(i) \cdot \frac{{}^{\mathbb{D}}n(i)}{{}^{\mathbb{D}}\pi} \tag{5.2} $$

CiteSeer submission model

We model the acquisition of papers in CiteSeer. We assume a simple direct submission of the document into CiteSeer database through a web form[5].

[5] http://citeseer.ist.psu.edu/submitDocument.html

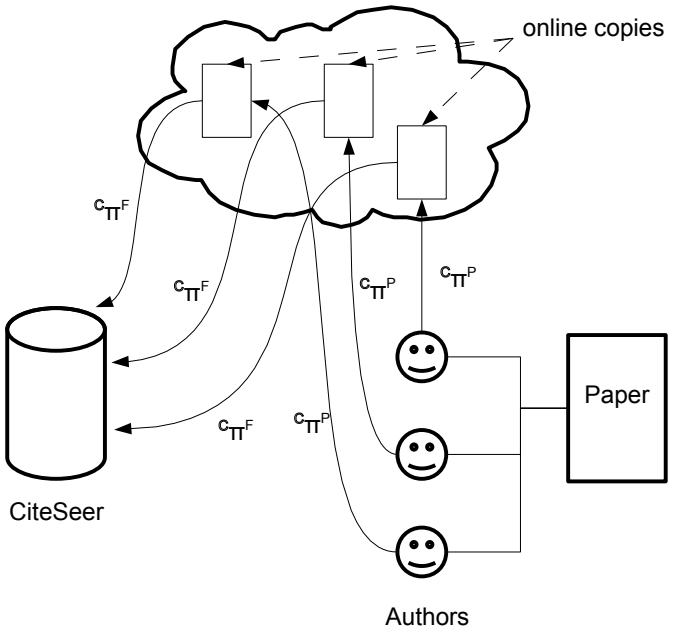


Figure 5.5: CiteSeer crawler acquisition model. Each author publishes a copy of the paper online with equal probability. Each copy has then the same chance of being found by the crawler.

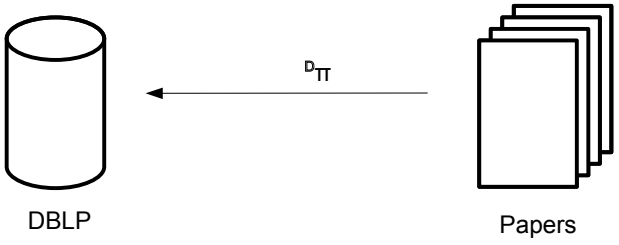


Figure 5.6: DBLP acquisition model. Each paper has the same probability of being included in DBLP.

For simplicity we assume that authors make the decision to submit or not independently and with equal probability. This *CiteSeer submission model* model is illustrated in Figure 5.4. Let ${}^{\mathbb{C}}\pi^S \in (0,1)$ be the probability that an author submits a paper directly to CiteSeer. Then ${}^{\mathbb{C}}\pi(i) = 1-(1-{}^{\mathbb{C}}\pi^S)^i$ where $(1-{}^{\mathbb{C}}\pi^S)^i$ is the probability that none of the i authors submit their paper to CiteSeer.

Substituting to (5.2) and re-arranging, we have

$$\mathrm{r}(i) = \frac{{}^{\mathbb{D}}n(i)}{{}^{\mathbb{C}}n(i)} = \frac{{}^{\mathbb{D}}\pi}{(1-(1-{}^{\mathbb{C}}\pi^S)^i)} \tag{5.3}$$

where $r(i)$ is the ratio of number of papers that have i authors in DBLP versus the number of papers with i authors in CiteSeer. It is clear from Equation 5.3 that as the number of authors, i, increases, the ratio, $\mathrm{r}(i)$, tends to ${}^{\mathbb{D}}\pi$, i.e. we expect that the number of i-authored papers in CiteSeer will approach all(i) and thus from Equation 5.1 the ratio tends to ${}^{\mathbb{D}}\pi$. For single authored papers, i.e. $i=1$, we have that $\mathrm{r}(1) = \frac{{}^{\mathbb{D}}\pi}{{}^{\mathbb{C}}\pi^S}$ and since we know that DBLP has more single-authored papers, it must be the case that ${}^{\mathbb{C}}\pi^S < {}^{\mathbb{D}}\pi$. More generally, we expect the ratio, $\mathrm{r}(i)$, to monotonically decrease with the number of authors, i, reaching an asymptote of ${}^{\mathbb{D}}\pi$ for large i. This is approximately observed in Figure 5.7, ignoring points for $i > 30$ for which there is not enough papers available. (There are few papers with such a high number of authors.)

In Figure 5.8 we plot the proportion $\mathrm{r}(i)$ for numbers of authors i where we have at least 100 papers available. We also display a fit of Equation (5.3) to the data in Figure 5.8. Note that Figure 5.8 contains the same data as Figure 5.7 but restricted to $i < 17$. We see the fit may partially explain the bias observed.

The value to which the data points are converging for high numbers of authors is ${}^{\mathbb{D}}\pi \approx 0.3$. We have to take into account that we only used 71% of DBLP papers in the dataset (out of total ${}^{\mathbb{D}}\mathrm{N}$ papers in DBLP dataset) and 57% of CiteSeer papers (out of total ${}^{\mathbb{C}}\mathrm{N}$ papers in CiteSeer dataset) in our analysis – the papers that have both year and number of authors specified. Substituting ${}^{\mathbb{D}}\pi \approx 0.3$ into (5.4) we get the value of ${}^{\mathbb{D}}\pi' \approx 0.24$. If our model is correct, this would suggest that the DBLP database covers approximately 24% of the entire Computer Science literature.

$${}^{\mathbb{D}}\pi' = \frac{{}^{\mathbb{D}}\mathrm{N}(i)}{{}^{\mathbb{C}}\mathrm{N}(i)} = \frac{0.57}{0.71} \cdot \frac{{}^{\mathbb{D}}n(i)}{{}^{\mathbb{C}}n(i)} = 0.8 \cdot {}^{\mathbb{D}}\pi \tag{5.4}$$

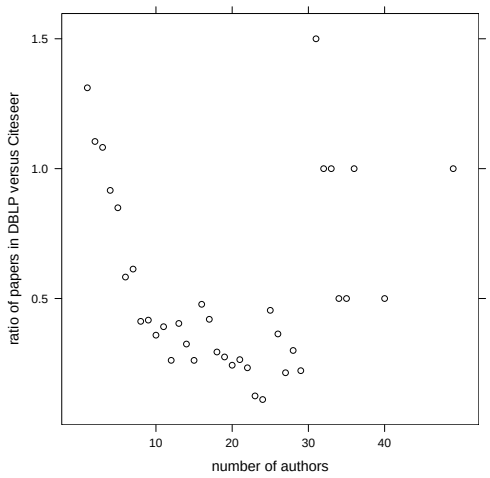


Figure 5.7: Ratio, $r(n_a)$ of the number of papers with n_a authors present in DBLP to the number of papers with n_a authors present in CiteSeer. The x axis represents the number of authors n_a and y-axis the ratio $r(n_a)$.

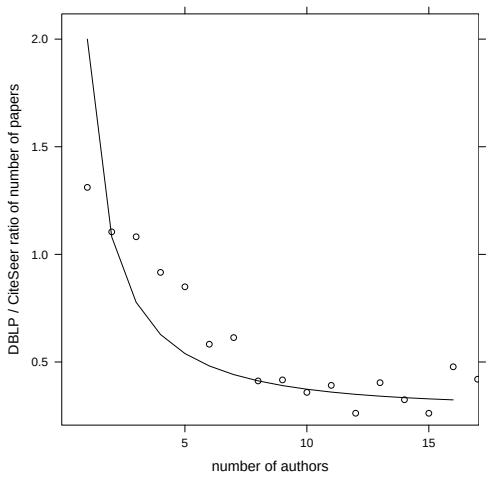


Figure 5.8: Fit of the submission model with parameters ${}^{\mathbb{D}}\pi = 0.3$ and ${}^{\mathbb{C}}\pi^{S} = 0.15$, Number of authors on x-axis is limited to cases where there are at least 100 documents in both datasets available.

CiteSeer Crawler Model

CiteSeer not only acquires papers based on direct submission by authors, but also by a crawl of the Web. We model this process in a *CiteSeer crawler model*

illustrated in Figure 5.5. We assume the probability of an author publishing a copy of the paper online (on her homepage for example) is independent and equal for all authors.

To begin, let ${}^{\mathbb{C}}\pi^{P} \in (0,1)$ be the probability that an author publishes a paper on a website. Then the average number of copies of an n_a-authored paper on the Web is $n_a \cdot {}^{\mathbb{C}}\pi^{P}$. Papers with multiple copies online are more likely to be found by the crawler. Let us further assume that the crawler finds each available on-line copy with a probability ${}^{\mathbb{C}}\pi^{F}$. If ${}^{\mathbb{C}}\pi^{P}(n_a, n_c)$ denotes the probability that there will be n_c copies of an an n_a-authored paper published on-line, then we can derive:

number of authors		probability of
1	${}^{\mathbb{C}}\pi^{P}(1,1) = {}^{\mathbb{C}}\pi^{P}$ ${}^{\mathbb{C}}\pi^{P}(1,0) = 1 - {}^{\mathbb{C}}\pi^{P}$	1 copy online 0 copies online
2	${}^{\mathbb{C}}\pi^{P}(2,2) = ({}^{\mathbb{C}}\pi^{P})^2$ ${}^{\mathbb{C}}\pi^{P}(2,1) = 2 \cdot {}^{\mathbb{C}}\pi^{P} \cdot (1 - {}^{\mathbb{C}}\pi^{P})$ ${}^{\mathbb{C}}\pi^{P}(2,0) = (1 - {}^{\mathbb{C}}\pi^{P})^2$	2 copies online 1 copy online 0 copies online
$\vdots$		
n_a	${}^{\mathbb{C}}\pi^{P}(n_a, n_c) = \binom{n_a}{n_c}({}^{\mathbb{C}}\pi^{P})^{n_c} \cdot (1 - {}^{\mathbb{C}}\pi^{P})^{n_a - n_c}$	n_c copies online of an n_a-authored paper

The total probability, ${}^{\mathbb{C}}\pi^{F}(n_c)$, of finding a document with n_c copies on-line, is

$$ {}^{\mathbb{C}}\pi^{F}(n_c) = 1 - (1 - {}^{\mathbb{C}}\pi^{F})^{n_c} \tag{5.5}$$

thus the probability that CiteSeer will crawl an n_a-authored document, ${}^{\mathbb{C}}\pi(n_a)$

is

$$
\begin{aligned}
{}^{\mathbb{C}}\pi(n_a) &= \sum_{n_c=0}^{n_a} {}^{\mathbb{C}}\pi^P(n_a, n_c) \cdot {}^{\mathbb{C}}\pi^F(n_c) \\
&= \sum_{n_c=0}^{n} {}^{\mathbb{C}}\pi^P(n_a, n_c) \cdot (1 - (1 - {}^{\mathbb{C}}\pi^F)^{n_c}) \\
&= \sum_{n_c=0}^{n_a} \left(\binom{n_a}{n_c} ({}^{\mathbb{C}}\pi^P)^{n_c} (1 - {}^{\mathbb{C}}\pi^P)^{n_a - n_c} \right) (1 - (1 - {}^{\mathbb{C}}\pi^F)^{n_c}) \\
&= 1 - \sum_{n_c=0}^{n_a} \left(\binom{n_a}{n_c} ({}^{\mathbb{C}}\pi^P)^{n_c} (1 - {}^{\mathbb{C}}\pi^P)^{n_a - n_c} \right) (1 - {}^{\mathbb{C}}\pi^F)^{n_c} \\
&\qquad \text{(sum of probabilities equals 1)} \\
&= 1 - \sum_{n_c=0}^{n_a} \left(\binom{n_a}{n_c} ((1 - {}^{\mathbb{C}}\pi^F) {}^{\mathbb{C}}\pi^P)^{n_c} (1 - {}^{\mathbb{C}}\pi^P)^{n_a - n_c} \right) \\
&= 1 - ({}^{\mathbb{C}}\pi^P (1 - {}^{\mathbb{C}}\pi^F) + (1 - {}^{\mathbb{C}}\pi^P))^{n_a} \qquad \text{(from binomial theorem)} \\
&= 1 - ({}^{\mathbb{C}}\pi^P - {}^{\mathbb{C}}\pi^F \cdot {}^{\mathbb{C}}\pi^P + 1 - {}^{\mathbb{C}}\pi^P)^{n_a} \\
&= 1 - (1 - {}^{\mathbb{C}}\pi^F \cdot {}^{\mathbb{C}}\pi^P)^{n_a} \qquad (5.6)
\end{aligned}
$$

where $(1 - {}^{\mathbb{C}}\pi^F \cdot {}^{\mathbb{C}}\pi^P)^{n_a}$ is the probability that no copy of an n_a-author paper is found by CiteSeer.

Once again, if we substitute Equation (5.6) in (5.2), we have

$$
\mathrm{r}(i) = \frac{{}^{\mathbb{D}}n(i)}{{}^{\mathbb{C}}n(i)} = \frac{{}^{\mathbb{D}}\pi}{(1 - (1 - {}^{\mathbb{C}}\pi^F \cdot {}^{\mathbb{C}}\pi^P)^i)} \qquad (5.7)
$$

which is equivalent to the "submission" model of Equation 5.3. That is, we have shown that both models lead to the same type of bias.

5.3.2 Citation network data analysis

After having inspected the attribute data $\mathcal{A}^A$ and $\mathcal{A}^Y$, we proceed with the analysis of the network data contained in CiteSeer and DBLP datasets. Citation distributions are one way of summarizing the citation data. Figure 5.9 compares citation distributions in CiteSeer versus DBLP. We see that DBLP contains more low cited papers than CiteSeer. This may be related to Lawrence's observation that articles freely available online are more highly cited [131]. CiteSeer contains by definition only papers available online compared to DBLP that has information on papers published in venues, some of which restrict free online publication of the full texts.

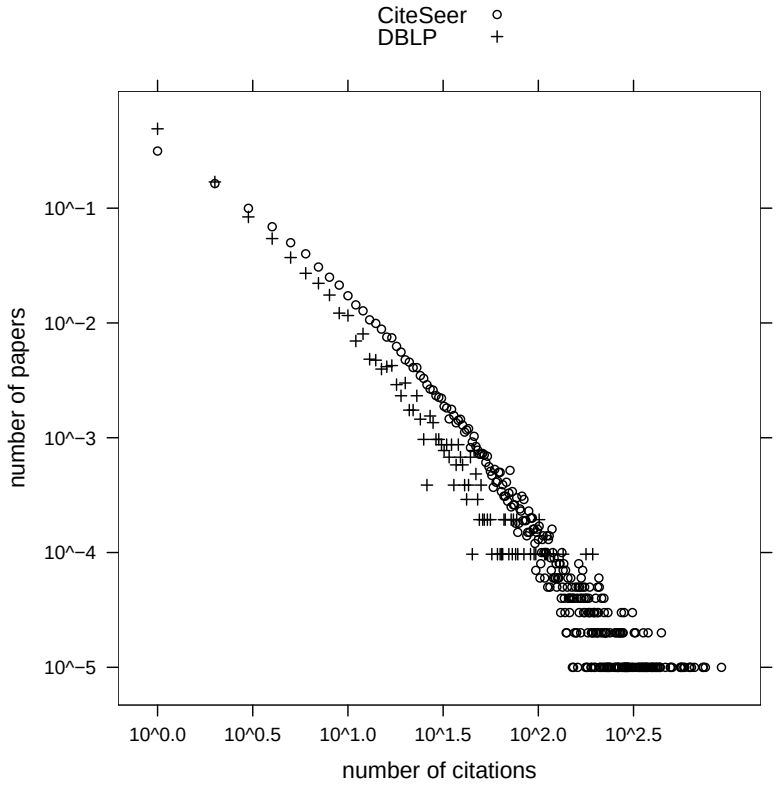


Figure 5.9: Atomic probability histograms on double logarithmic scales for number of citations in CiteSeer and DBLP. The y-axis represents the number of papers that receive a particular number of citations (x-axis).

The data is very noisy for high number of citations as highly cited papers are relatively rare. We use exponential binning (Figure 5.10) to estimate the parameters of the citation distribution in CiteSeer and DBLP. Exponential binning is a technique where the data are aggregated in exponentially increasing 'bins'. In this manner we obtain a higher number of samples in each bin, which leads to reduction of the noise in the data. If data is unevenly distributed variable sized bins can capture sets of data with comparable number of datapoints. The resulting average in each bin is therefore less noisy. As papers with higher number of citations are more rare, bins of exponentially increasing size work well.

We estimated the power-law parameters for CiteSeer and DBLP. The papers in each dataset have been first divided into two groups – papers with more than and less than 50 citations. We then performed a linear fit in each group separately. The slopes in Table 5.3 correspond to linear interpolation

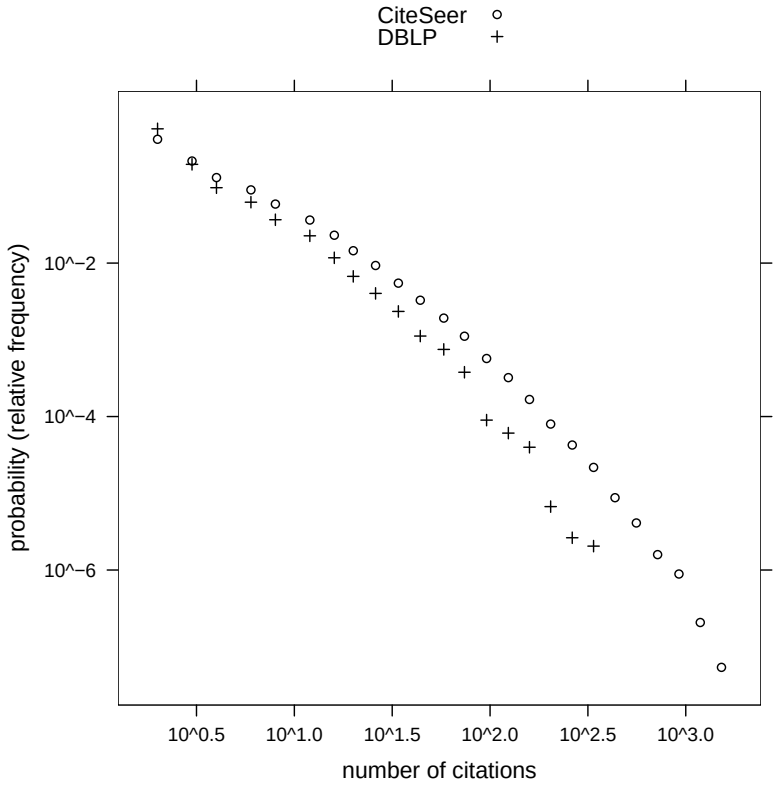


Figure 5.10: Exponentially binned probability histograms on double logarithmic scales for number of citations in CiteSeer and DBLP datasets.

of the exponentially binned data as displayed in Figure 5.10. Higher slopes in our datasets compared to those of Lehmann [134] indicate a more uneven distribution of citations.

For both datasets we obtain parameters bigger in absolute value than Lehmann [134] derived for Physics. This means that highly cited papers acquire a larger share of citations in Computer Science than in Physics.

5.4 Summary

This chapter presented an analysis of two popular online science citation databases, DBLP and CiteSeer. These two citation databases have very different methods of data acquisition. We showed that autonomous acquisition by web crawling, CiteSeer, introduces a significant bias against papers with low number of authors (less than 4). Single author papers appear to be dis-

number of citations	slope		
	Lehmann	CiteSeer	DBLP
< 50	-1.29	-1.504	-1.876
> 50	-2.32	-3.074	-3.509

Table 5.3: Citation distribution parameters for CiteSeer and DBLP. Slopes from Figure 5.10 representing the parameters of the corresponding power-laws.

advantaged with regard to the CiteSeer acquisition method. As such, single authors, who care, will need more actively submit their papers to CiteSeer if this bias is to be reduced.

We attempted to model this bias by constructing two probabilistic models for paper acquisition in CiteSeer. The first model assumes the probability that a paper will be submitted is proportional to the number of authors of the paper. The second model assumes that the probability of crawling a paper is proportional to the number of online copies of the paper and that the number of online copies is again proportional to the number of authors. Both models are, as we demonstrated, equivalent and permit us to estimate that the coverage of DBLP is approximately 24% of the entire Computer Science literature.

We then examined the citation distributions for both CiteSeer and DBLP and observed that CiteSeer has a small number of low-cited papers. The citation distributions were compared with prior work by Lehmann [134], who examined datasets from the Physics community. While the CiteSeer and DBLP distributions are different, both datasets exhibit steeper slopes than SPIRES HEP dataset, indicating that highly cited papers in Computer Science receive a larger citation share than in Physics.

We have seen that citation network properties vary across scientific disciplines and that acquisition methods may introduce significant bias in the way we perceive these networks. In the next chapter we study a different type of networks - the social networks as defined by friendship between users.

Chapter 6

Website navigation analysis

In this chapter we describe a user study we performed to evaluate three e-government websites in terms of navigability and information accessibility. We also discuss the structural properties of these websites and their relationship to user performance in simple tasks of locating relevant information. Some results presented in this chapter have been published in [177] and [66].

Most advanced industrial nations have put considerable political support and financial resources behind the development of e-government. By 2005, the UK for example has a '.gov' domain of around 8 to 23 million pages (depending on which search engine estimates one tends to believe, MSN or Google respectively) and was spending £ 14.5 billion a year on information technology in the pursuit of the Prime Minister's commitment to have all government services electronically available by the end of 2005. In spite of these resources (more than 1 per cent of GDP in most industrialized nations is spent on government information technology), e-government tends to lag behind e-commerce. In the UK, recent survey evidence [63] suggests that while 85 percent of Internet users claim to have looked for or bought goods and services online, and 50 percent of users to shop online at least once a month, only 39 percent have had any sort of interaction with government online in the last year. While figures for e-government usage are much higher in some countries, particularly Scandinavian, the generalization that government has been far less touched than commerce by widespread use of the World Wide Web holds true internationally. Governments are under pressure to demonstrate that the massive investments they are making are worthwhile. There have been extensive efforts to assess the quality of e-government throughout the world. An overview of this work is provided in Section 6.1. Similarly, there have been numerous studies within the computer science community to assess and characterize the structure of hyperlinked en-

vironments, as we have discussed in Section 4.1. Despite the amount of work in webmetrics and e-government evaluation, these two have rarely been combined. We compare three different e-government websites in a user based experiment and contrast the results with metrics computed for the structure of their link graphs. Metrics informative about the quality of user experience may be used to assess websites, possible improvements to their structure and evaluate the investments made during their development.

6.1 Previous e-government studies

There have been numerous attempts to assess e-government internationally. These generally have the form of rankings of countries carried out or commissioned by international organizations (such as UNPAN [214], European Commission [73]), private sector consultancies (particularly by Accenture [4], Taylor Nelson Sofres [205] and Graafland-Essers and Ettedgui [83]), and academic commentators [63, 226, 180, 41]). While some are widely cited and eagerly awaited by governments which score well, many are of methodological questionability and rely, ultimately, on subjective judgments. Most make some form of assessment of government websites according to content (eg. [226]) and availability of services (eg. [73]). Accenture's widely known annual study is largely a qualitative analysis. It is based on researchers' assessments of websites, e-services available on them and a limited number of short visits of researchers to the 22 countries covered. All these studies fail to collect either user-based or structural metrics. None have been able to collect user data (access log files for example) for significant numbers of websites, especially due to the cooperation problems with individual website administrators. Some studies use survey evidence to estimate the extent to which a population as a whole have interacted with their government online (Taylor Nelson Sofres [205] in particular, while Accenture included a user opinion survey for the first time in 2005 [4]). West [226] gathered content-related data from approximately 2,000 websites in nearly 200 countries and LaPorte and Demchak developed measures of 'interactivity' and 'transparency' for tracking the diffusion and use of the Web in nearly 200 governments around the world (now discontinued). However, none of these studies have considered navigation and the structure of links between pages on e-government websites. Methodological variations across these studies are evidenced by the different rankings that the countries achieve.

Usually, the study of hyperlink structure has focused on academic networks [37, 210]. Outside academia there have been many efforts concerned with website usability from a user perspective [156]. Many books offer recom-

mendations on good design, accessibility for disabled users, etc, for example Nielsen [156].

Content and presentation also may have a big effect on users judgment of the website. Ivory et al. [109] compared a set of content based metrics such as length of pages, graphics to text ratio, fonts etc to the manual evaluation of websites by human experts. They showed that the content based metrics have a reasonable predictive capability for the quality of the website, as judged by human experts, and also that the same feature may have positive impact in one class of websites while it may be a negative sign for a different class of websites.

The application of computer science methods to the study of politics on the web and e-government in particular is not yet very common, although there are some notable exceptions. For example, Hindman et al [99] studied the communities surrounding political websites and showed that i) the number of incoming links is highly correlated with the number of actual users and ii) that online communities are usually dominated by a few websites – winners who take all the attention. Overall, applications of structural metrics and lab experiments to the quantitative evaluation of e-government have not been reported. Same holds for user studies in controlled environment.

6.2 Website datasets

We have selected the foreign office (FO) websites of Australia (AU), the United Kingdom (UK) and the United States (US) for this study. These departments were chosen because they are all targeted at an English-speaking audience and have roughly comparable roles across the countries.

Defining the borders of a website is not straightforward. The foreign offices distribute their services across a number of domains, most notably providing separate domains for visa and passport services and travel advice. We performed the identification of pages belonging to each website manually. To decide which pages to include we asked the following two questions:

1. Is the service offered supposed to be an FO responsibility?
2. Is the website operated by the FO?

We included pages for which the answer to both of the question was yes. Table 6.1 gives an overview of the departments and the corresponding domains that we used in our study. We excluded embassy websites as they are usually operated rather independently from the foreign office website and generally indicated by another domain. Only some US embassies are hosted directly under the state.gov domain.

	Department name	Domains
AU	Department for Foreign Affairs and Trade	http://www.dfat.gov.au http://www.smartraveller.gov.au http://www.passports.gov.au http://www.trademinister.gov.au http://www.foreignminister.gov.au
UK	Foreign and Commonwealth Office	http://www.fco.gov.uk http://www.ukvisas.gov.uk
US	State Department	http://www.state.gov http://www.unitedstatesvisas.gov

Table 6.1: Departments and corresponding websites included in our sample.

During December 2006 we collected the link structure of the three foreign office websites by performing an exhaustive breadth-first crawl. Each crawl was started from the homepage of the respective website and was restricted to the domains specified in Table 6.1. An exhaustive crawl was performed with a security limit on the depth of 18. In addition we manually identified and blacklisted spider traps including a mailing list archive and a diary application on the US website. The crawler extracted links from HTML documents but did not parse other content (*.pdf, *.doc, etc). This should not introduce any significant bias as links in the content not parsed form a negligible fraction of links that are actually followed by users. As we crawled a small number of large websites we had to limit the rate of our queries in order not to turn our crawling into a denial-of-service attack. The crawler was configured to run a single thread with 5s delay between requests and to store all links between pages including the links internal to the websites.[1] We used the opensource crawler Nutch version 0.6, which we modified to use the HTTP/1.1 protocol[2]. We ran the crawls on a PC with Intel Pentium 4 processor at 2.80GHz, 1G RAM and over 1TB networked disk storage. Neither hardware nor software were a bottleneck. Once collected, we dumped the data in Pajek format [15] for further analysis. At the end the data contained 32K pages and 895K links for AU, 24K pages and 430K links for UK and 129K pages and 2.5M links for US (Table 6.2).

[1] A link is called internal when both the origin and the destination page are part of the same website – share a domain name. Such links are often ignored by crawlers as they are not used for PageRank computation.

[2] The Lotus Domino servers, very popular with government websites, incorrectly handle zero content-length HTTP/1.0 requests and return HTTP Error 400 "Bad Request". This prevents Nutch v0.6 from crawling the website completely. By patching Nutch to use HTTP/1.1 we were able to perform an exhaustive crawl.

	number of pages			number of links
	whole website	HTML	doc/pdf/etc.	
AU	32,765	30,690	1,667	895K
UK	23,570	20,867	2,439	430K
US	129,246	115,888	13,091	2.5M

Table 6.2: Foreign office websites. The number of pages and links crawled.

6.3 User experiment design

The goal of the experiment was to compare the three foreign office websites of Australia, United Kingdom and United States and to observe the browsing behavior of subjects in a controlled environment of a computer lab. We are particularly interested in whether subjects would find information easier by using external search engines, internal search capabilities or by direct navigation of the e-government websites.

To evaluate the websites we asked participants to locate information that we knew was present there and pay them by performance. Every subject received a flat rate of £5 for taking part in the study and a further £0.50 for each correctly answered question. This motivated the subjects to answer as many questions as possible.

Having motivated the users by sufficient financial compensation, we considered several metrics for user performance: number of questions answered, number of questions correctly answered, number of correctly answered questions per time unit etc. We decided to use *success-rate* which is defined as the number of correctly answered questions per minute (considering the time taken to answer all correctly and incorrectly answered questions). The success-rate measure rewards websites on which the answers are easy and quick to find. We believe this metric reflects well the utility of users in real situations. Unlike to other types of websites, users come to the foreign office websites to find answers to their questions, not to kill time or be entertained.

We used a 3x3 design where we tested the three websites AU, UK, US in three treatments 1, 2 and 3 which resulted in nine distinct groups of subjects. The three treatments reflected three different scenarios – 1) unrestricted surfing of the Web, 2) searching the foreign office website directly and 3) navigating the foreign office website with internal search disabled.

Treatment 1 – users were presented with a blank page in a browser and were allowed to use the whole Web to locate the information. This treatment is the one that is the most close to real situation where users are allowed to use tools of their choice. We were curious if the subjects

will use government websites at all and what competing information providers they would find.

Treatment 2 – users started on the home page of the corresponding foreign office and their browsing was restricted to the foreign office website only. We were curious what will be the usage of websites internal search box and the success rate of answering questions when we knew the answers were present within the website.

Treatment 3 – users were further restricted compared to Treatment 2 by the fact that the internal search facility on the foreign office websites was disabled. Again users could use only the foreign office website but this time they had to navigate to the final page by clicking on hyperlinks only. We were curious to see how long subjects would take to find answers to our questions and what represents a more efficient strategy on these websites - searching or navigation?

In order to test how easy or difficult it is for people to find information from foreign office websites, we prepared a number of questions that asked for particular information provided by the three websites. The questions covered information generally accepted as responsibility of foreign offices in the three countries. Users were asked to find, for example, travel advice or information about obtaining a visa. In reality, each of these questions is of different importance and the answer is sought with different frequency. Still we believe that a good foreign office website would provide the information we requested faster than a poorly designed website and our approach provides a good substitute for usage data which is not available. We asked participant 16 questions which comprised of 10 questions that that have been confirmed by two political scientists and also by our subjects to cover the information that citizens seek and expect to find on a foreign office website. The remaining 6 questions were targeted at information that is incidentally found on the foreign office websites but which is not essential to the foreign office mission, such as the birth day of an ambassador for example. In the analysis presented in this chapter we used only the 10 questions covering the essential information that a foreign office has to provide as this is what matters. For the full list of the ten questions used please see Appendix A.1.

The participants of our experiment consisted of a self-selected sample of 134 person from various backgrounds but most of them were students. The requirements for participation in this experiment were basic computer skills. The majority of our participants were aged between 20 and 25 years and regular users of the Internet. Recruitment took place through the subject database of the ESRC Centre for Economic Learning and Social Evolution.

The database consists of volunteers that have indicated that they would be available for experiments. Potential subjects were contacted via email and were randomly allocated to one of the sessions on positive reply.

Because of the scale of the experiment, testing in the laboratory happened during three days in January 2006 in 4 sessions per day. For each session, participants were randomly distributed across groups while preserving approximately equal numbers in each group. We did this to control for the time-of-day effects (for example: users are more sleepy early in the morning, immediately after meals and late in the evening than at other times during the day). If this even distribution was not observed, differences in performance could be dominated by the time of day effect as opposed to experimental setting.

During the experiments, subjects were seated so that they could not collaborate and were constantly monitored. For the experiment, each subject was assigned a unique anonymous identifier. This identifier determined the group (country and treatment) and subjects used it to log into the online experiment interface.

Each session followed identical plan depicted in Figure 6.1. At the beginning of each session participants received an oral introduction covering the purpose and organization of the experiment and the amount of compensation they could expect. Users then read written instructions, and proceeded to answer the questions paid by performance. The participants had 45 minutes to answer all questions. The interface would inform them about the time they have taken so far, the total as well as the remaining number of questions (see screenshot in Figure 6.2). At the end of the time limit or after all 16 questions had been answered, the participants were asked to fill in a post-questionnaire and provide feedback on the experiment.

The interface used Firefox web browser and provided two windows: one holding the questions to be answered and one to be used for surfing the web in order to locate the information for each question. Subjects could not return to a question once it was answered or skipped. Participants were asked to select the correct answer and to provide the URL of the page on which the information was found.

Figure 6.3 shows the technical setup of the lab during user experiment, which ensured that subjects were able to access only the pages allowed in their group. Access of users was controlled centrally from the proxy server. All browsers had identical setup with no bookmarks and a slogger[3] plugin installed. Slogger was used to keep track of users' browsing history including the use of back button to browse cached copies of a web page – this data would

[3] https://addons.mozilla.org/en-US/firefox/addon/143

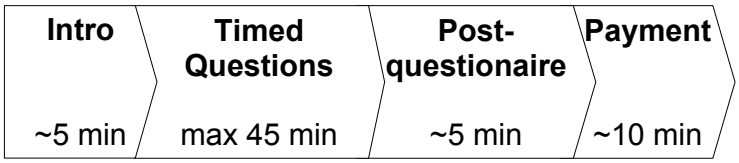


Figure 6.1: Flowchart of the user experiment time plan.

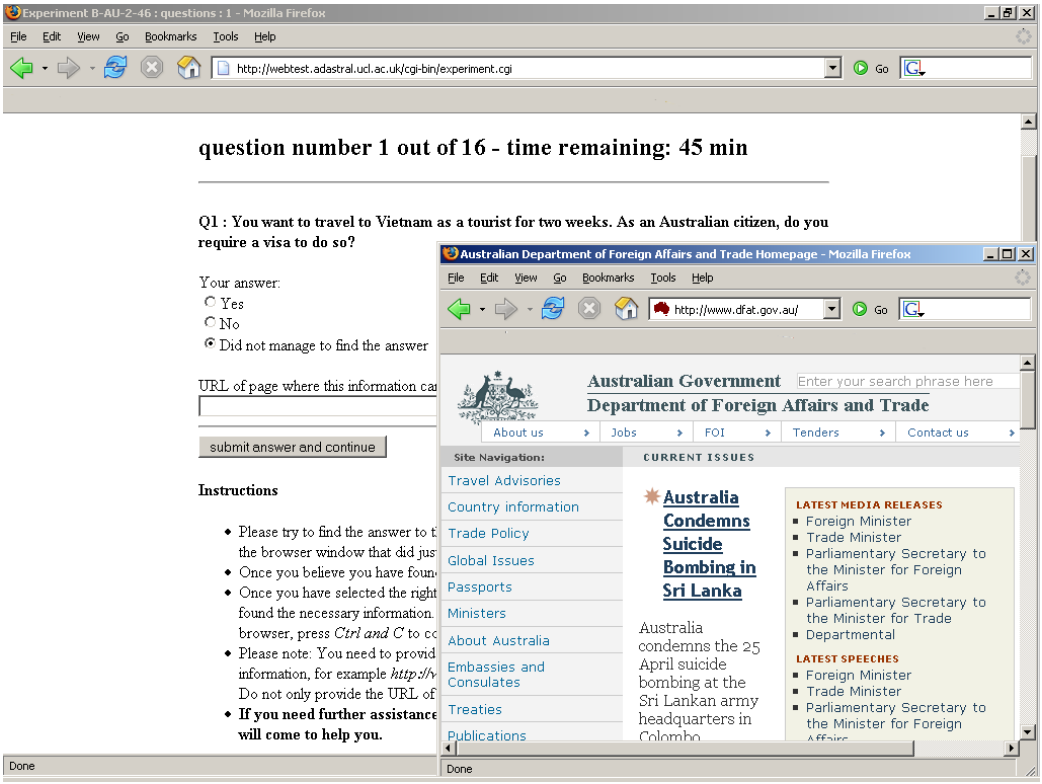


Figure 6.2: Screenshot of experiment interface, displaying one window with the question and another (resizable) window that could be used to surf the Internet to find the relevant information for answering the question.

not be available on the proxy server as the browser does not communicate with the proxy server if the data requested is already in its cache.

During the study we recorded the answers to the multiple choice questions, the URL of the page containing the answer, timestamps of events during the whole experiment including each answer, the content of the questionnaire, and also the complete surfing history of subjects. We analyze this

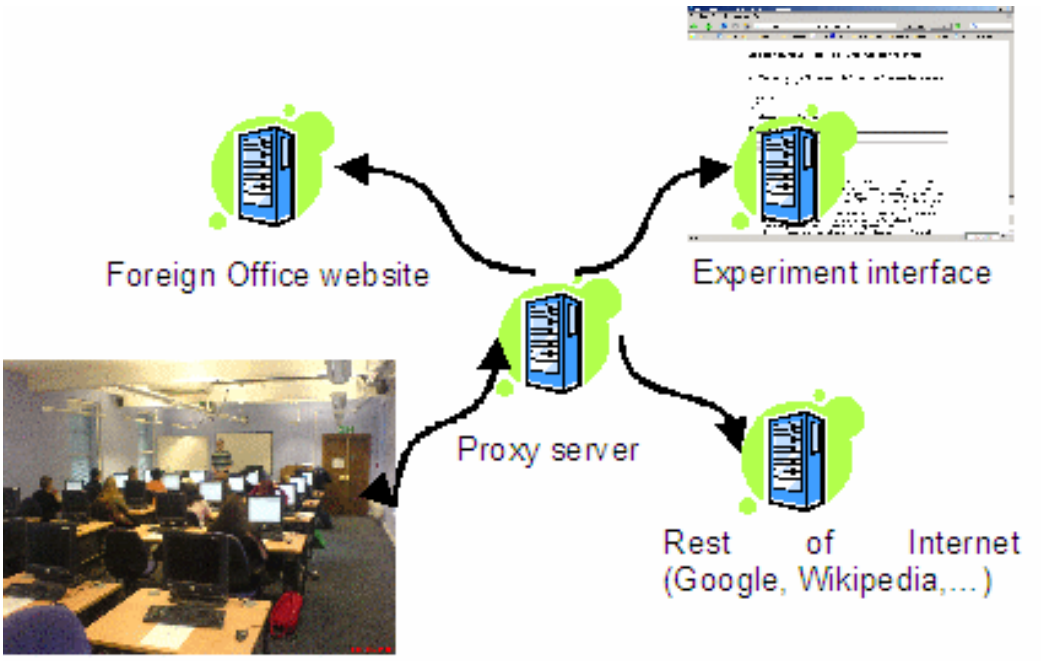


Figure 6.3: Technical setup of the user experiment.

data in the next section.

6.4 Results

In this section we first present results of the user experiment, then we describe properties of the website link structure and finally discuss the relationships between the structural properties and user performance in our experiment.

6.4.1 User experiment results

	Treatment 1	**Treatment 2**	**Treatment 3**	
	unrestricted	website	website navigation	sums
AU	15	18	12	44
UK	15	19	11	45
US	15	18	12	45
sums	45	55	34	134

Table 6.3: Number of users in each group for the user experiment.

	AU	**UK**	**US**
Treatment 1	0.401 (0.12)	0.438 (0.13)	0.445 (0.09)
Treatment 2	0.418 (0.14)	0.411 (0.09)	0.294 (0.06)
Treatment 3	0.301 (0.08)	0.458 (0.11)	0.260 (0.09)

Table 6.4: Success rate of user groups. Value reported in parentheses is standard deviation.

We had in total 134 participants. Table 6.3 gives an overview of the actual number of subjects in the nine groups. The 134 users were divided between countries AU, UK and US with each country having 44, 45 and 45 users respectively. In each treatment we then had 45, 55 and 34 users respectively. We measured their performance using success-rate metric defined earlier. Table 6.4 shows the performance of the different groups with standard deviation within these groups and Figure 6.4 displays this performance together with 95% confidence intervals. As we are going to compare countries within treatments and then each country across treatments we are going to test relatively high number of hypotheses. To avoid false identification problem, we adjust the confidence level of individual tests by Bonferroni correction [3]. In total we will test 18 hypotheses – three pairwise comparisons between countries within each treatment and three pairwise comparisons between treatments for each individual country. To achieve overall 95% confidence the Bonferroni adjustment gives us individual confidence levels of $(1 - \frac{(1-0.95)}{18}) \approx 0.9972$. All 18 pairwise tests on difference in mean success-rate have been performed at the 99.72% confidence level. We use Welch's t test which does not assume equal variance of the two samples. In addition, we also used an alternative approach to multiple hypothesis testing, the Tukey honest significant differences procedure (HSD) [67]. HSD also confirmed the significance of the tests discussed below and we present the full output of this procedure in Appendix A.2. The differences between groups remained significant even after taking into account the self-reported Internet skills of subjects as elicited in the experiment questionnaire. Below we first discuss the individual treatments separately, and then the differences between the treatments.

Treatment 1

In treatment 1, we simulated the usual situation of citizens. We were interested in three aspects of their information seeking i) what means the subjects will use to find information (global search, website internal search, navigation), ii) where will they find the information (government or independent websites) and iii) for which country the information is more accessible.

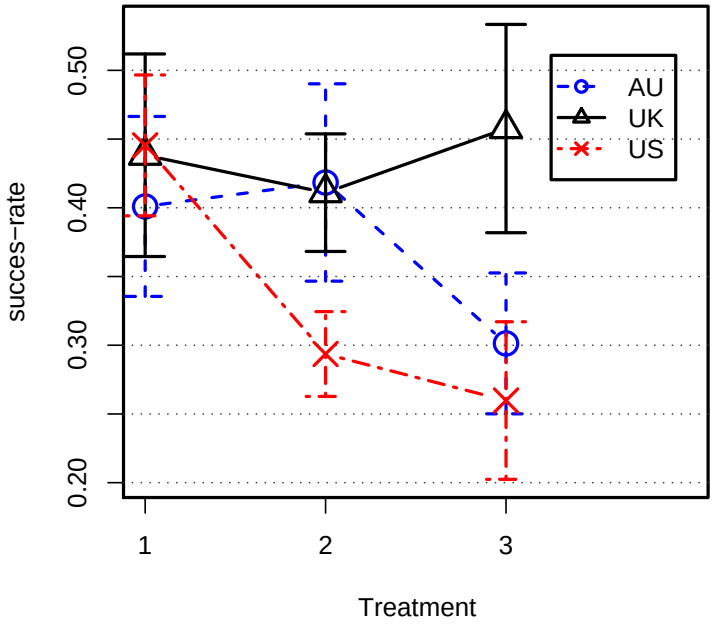


Figure 6.4: Success-rate (Average number of correctly answered questions per minute) by treatment and country with confidence intervals ($p = 0.56$) – this level has been chosen so that two confidence intervals are disjoint if and only if the two random variables have different mean with 95% confidence.

There are two types of search engines that differ in coverage and often performance. Global search engines (*GSE*) such as Google, Yahoo, MSN and others index substantial parts of the Web. Many websites provide internal search engine (*ISE*) that is indexing the content of the website only. The advantage of an *ISE* is that it may be tailored to suit specific needs, type of content and may be able to index documents not accessible to *GSEs* – documents in a database for example.

In Table 6.5 we can see that virtually all users used a *GSE* at least once and that the majority of the questions was answered using a *GSE*. Users often resorted to using the *ISE* if they did not find the information on the page. Despite the proliferation of searching, users still navigated extensively on the websites where they landed. Over 80% of questions were answered without the use of *ISE*.

	AU	**UK**	**US**
users who used *GSE* at least once	87%	100%	100%
questions answered using *GSE*	61%	75%	80%
users who used *ISE* at least once	80%	60%	53%
questions answered using *ISE*	16%	13%	19%

Table 6.5: The means by which subjects in Treatment 1 found answers.

	AU	**UK**	**US**
questions answered using the foreign office website	70% (0.1)	53% (0.2)	58% (0.1)
questions answered using government website of the country	83% (0.1)	73% (0.1)	80% (0.1)
questions answered using any government website	93% (0.1)	84% (0.1)	90% (0.1)
questions answered using non-government website	7% (0.1)	16% (0.1)	10% (0.1)

Table 6.6: The sources where subjects in Treatment 1 found answers.

Government websites formed a large proportion, around 90%, of the pages where subjects found the information (see Table 6.6), although some of the websites (approximately 10% for all three websites) were of government from a different country than the primary source. Between 7 and 16% of questions have been answered using information provided on websites out of the control of any government such as commercial travel websites or Wikipedia. Only in 60% of the cases the information was found on the foreign office website which is supposed to be the authoritative source for this type of information.

There is less competition from private information providers than from governments itself. This duplicit sources of information on other government websites are an double edged sword – they make it easier to find the information, but, on the other hand, may lead to maintenance issues when the information changes and needs to be updated.

There are differences between countries in terms of the fraction of questions answered using the foreign office website. Significantly more people used the AU foreign office website to answer the questions, indicating that there is less competition of information providers for AU than for the other two countries, or, in other words, that AU has a better relative visibility on search engines compared to competing websites.

Although the US outperformed UK which in turn outperformed AU in terms of success-rate in treatment 1, the absolute values are very close and

	AU	**UK**	**US**
internal search used at least once	72% (0.1)	95% (0.1)	94% (0.1)
average use of internal search (as %age of questions for which it was used)	30% (0.3)	39% (0.3)	35% (0.2)

Table 6.7: Search usage versus navigation in treatment 2. Value reported in parentheses is standard deviation where 0.1 corresponds to 10%.

the differences between them are not statistically significant. All three groups performed similar when when users are free to choose how and where to find the information.

Treatment 2

In treatment 2, users were starting their search from the home page of the respective foreign office website and were limited to this website only. They were allowed to use whatever facilities on the website there are to locate the information.

As we see in Table 6.7, between 72% and 94% of users tried at least once to use the internal search engine but overall subjects used internal search on less than 40% of questions and over 60% of questions have been answered using navigation only. Interestingly, the use the internal search is negatively correlated with the speed at which users found answers. While this might seem to suggest that *ISE* actually slows users down, it is more likely that subjects in trouble (subjects with less experience or those simply stuck) resort to search while subjects with more experience, or ideas where to look, would navigate. The assumption that users resorted to *ISE* when they did not know where to look only and otherwise followed links directly is supported by the comparison between treatment 2 and treatment 3 later on.

On the AU website, subjects used the internal search engine significantly less (72% in Table 6.7) than on US and UK where nearly all users tried it. The number of questions answered using internal search engine is very similar for all websites. In Table 6.4 and Figure 6.4 we see that subjects on US website performed worse than those on AU and UK. The pairwise Welch's t test confirms statistical significance of this difference. There is no statistically significant difference between the AU and UK. Subjects on the US website needed close to a minute more per question. Subjects using AU and UK websites would correctly answer 10 percent more questions than the subjects using the US website.

Treatment 3

Treatment 3 is the scenario forcing subjects to navigate by following links. In this treatment we blocked access to the internal search facility. We were curious if users will be able to find the information or if the search engines are indispensable.

The observed order of countries according to decreasing success-rate is UK, AU, US. Performance of the subjects on the UK website were not affected by the restrictions at all. We even recorded the highest absolute success-rate of all treatments, and groups (0.445 q/min). In this treatment there is no significant difference between AU and US which both perform poorly compared to UK – the advantage of UK is statistically significant at the 99.72% confidence level. Subjects on the UK websites performed on average twice as well as those on the other two websites. The websites themselves account for about 44% of the variation in user success-rate.

Differences between treatments

This discussion again refers to Figure 6.4 which also shows how success-rate is affected by treatment. We performed nine pairwise tests - three for each country to assess if the differences between treatments are statistically significant. Four of these pairwise tests are statistically significant at 95% overall significance level.

AU subjects performed as well when they used *GSE* (Treatment 1) as when they started directly on the website itself (Treatment 2). Once the internal search of the website was prohibited in Treatment 3, the success-rate significantly dropped. This suggests that some of the answers are very hard to find on the AU website using navigation only. For this website *GSE* or *ISE* is necessary to find all answers efficiently.

UK website performed as well in all treatments and the small absolute differences in success-rate are not statistically significant. This website was the top performer in all treatments and was surprisingly unaffected by the various restrictions.

US website performed as well as other countries in unrestricted treatment 1 but the performance degrades significantly when subjects cannot benefit from the *GSE*. This website is too big to be navigated efficiently and the internal search engine does not help much.

There is a general trend according to which the performance decreases with increasing restrictions – overall success-rate dropped from treatment 1 through treatment 2 to treatment 3.

Country	pages	Directed Distance			average degree
		average	median	maximum*	
AU	32,765	8.1	6	38	54.6
UK	23,570	4.9	5	10	36.5
US	129,246	6.2	6	17	38.8

Table 6.8: Structural properties of the three foreign office websites. (* diameter)

6.4.2 Link structure properties

In this section we analyze the structure of the three foreign office websites and report several structural metrics that may be related to the websites' navigability. The users follow links to get closer to the target page and to find pages related to the one they are currently viewing. The probability of a user following a particular link is influenced by the presentation of the link on the web page such as its position, color, font size, anchor text and context. In this work we abstract from the content of web pages and its presentation and focus solely on the link structure. Such abstraction has been common in web analysis [28, 165, 120].

In Table 6.8 we see that the three foreign office websites have very different structural characteristics. Directed distance characterizes the interconnectedness of the link structures. Lower average directed distance corresponds to the expected distance between two random pages. Intuitively it is beneficial to users to reach other pages inside the same website in a small number of clicks. We see that although the size of the websites is big, the average distance between random pages is relatively small. Still, a user viewing a page on the Australia website may need to perform up to 38 clicks to get to another page within the same website. It is reasonable to assume that most of the users would in such a case resort to other means than navigation and perhaps leave the website without seeing the other page.

Table 6.8 also demonstrates that bigger websites do not necessarily have a bigger average distance or diameter (compare AU and US for example). Diameter of the website may be reduced by addition of links and increasing the average degree of a page. This needs to be done with caution as an overload of links on one single page will only confuse users, increase probability of a mistake and decrease their navigational performance.

We also report the size of the bow-tie components of the link graphs in Table 6.9. These components give an idea of how restricted users may be in their navigation if they land in different parts of the website. Due to the nature of the crawl, which was started from the single website top page which

is part of the LSCC, the OUT component, DISCONNECTED components and TENDRILs are all empty. Interestingly, the size of the OUT component is equal to the percentage of unreachable pairs of pages in the network. This is because there are almost no links between pages in the OUT component resulting in these pages to be unreachable from each other. We can then show the following: Let us denote $|FO|$ the number of pages on the website, RP the set of reachable pairs of pages, UP the set of unreachable pairs of pages and AP the set of all pairs of pages. Then we have:

$$|AP| = |RP| + |UP| = |FO|^2 = (|LSCC| + |OUT|)^2 \tag{6.1}$$

as all other components are empty. Reachable pairs are then such where both pages are from $LSCC$ or the starting page is in $LSCC$ and the other page in OUT component. The count of reachable pairs is then:

$$|RP| \approx |LSCC|^2 + |LSCC| \cdot |OUT| \tag{6.2}$$

On the other hand, the number of unreachable pairs may be computed as the number of all possible pairs between pages in OUT, as virtually all of these are unreachable and the number of pairs between a starting page in OUT and target page in $LSCC$ – there is no path between such pages otherwise the page in OUT would have to be part of the $LSCC$.

$$|UP| \approx |OUT|^2 + (|OUT| \cdot |LSCC|) \tag{6.3}$$

The fraction of unreachable pairs is then

$$\frac{|UP|}{|AP|} \approx \frac{|OUT|^2 + |OUT| \cdot |LSCC|}{|FO|} \tag{6.4}$$

$$\approx \frac{|OUT|^2 + |OUT| \cdot |LSCC|}{|OUT|^2 + |OUT| \cdot |LSCC| + |LSCC|^2 + |LSCC| \cdot |OUT|} \tag{6.5}$$

$$\approx \frac{|OUT|^2 + |OUT| \cdot |LSCC|}{|OUT|^2 + |OUT| \cdot |LSCC| + |LSCC|^2 + |LSCC| \cdot |OUT|} \tag{6.6}$$

$$\approx \frac{|OUT|^2 + |OUT| \cdot |LSCC|}{(|OUT| + |LSCC|)^2} \tag{6.7}$$

$$\approx \frac{|OUT|}{|FO|} \tag{6.8}$$

And so we have the fraction of unreachable pairs equal to the fraction of pages in OUT component and consequently the fraction of reachable pairs equal to the number of pages in the $LSCC$ component:

$$\frac{|RP|}{|AP|} \approx \frac{|LSCC|}{|FO|} \tag{6.9}$$

Country	*LSCC*	*OUT*	reachable pairs	unreachable pairs
AU	89%	11%	89%	11%
UK	65%	35%	65%	35%
US	75%	25%	75%	25%

Table 6.9: Bow-tie structure of the foreign office websites. *IN*, *TUBE*, *DISCONNECTED* and *TENDRIL* components are all empty

The definition of the bow-tie structure is based on the existence of paths between pages. The actual distance between pages is also important. While a large *LSCC* is intuitively a good thing, it may be important how many clicks a user needs to reach a particular page in the *LSCC*. There may be a path but if it is too long, users may not find it, or give up before reaching the target page. Similar to Huberman et al. [106] we observe that users navigate over paths with limited length and we have seen in the experiment that on average, if users find the target page, it is generally in less than six clicks. Given the long tail distribution of path lengths, most of the paths are shorter than average.

To compare the structure of the website with respect to accessibility of pages as a function of depth of navigation we plot the reachability of the link graph in Figure 6.5 and the reachability from the home page of the website in Figure 6.6. These figures display the cumulative distribution of the fraction of pages that are accessible from a random page and from the home page respectively.

Figure 6.5 corresponds to the situation when a user lands within the website from search engine, or maybe just finished looking for some unrelated information on this website and now needs to navigate to a different page. The plot was constructed in the following way. We computed the distance for all possible pairs of pages and then displayed the percentage of the path-lengths that are shorter than x. Not all pairs of pages have a path between them and so the cumulative percentage converges to the fraction of reachable pairs which is less than 100% for the three websites (89% for AU, 65% for UK and 75% for US). Interestingly, the *LSCC* of AU is impressive 89%, but users still need many clicks to actually get from one page to another. While UK has the smallest *LSCC* it is the only website out of the three where more than half of the content is accessible within six clicks.

Figure 6.6 corresponds to the situation where the user starts from home page and tries to navigate to a random page within the website. For the UK, almost all pages are within a distance of six clicks from the homepage and

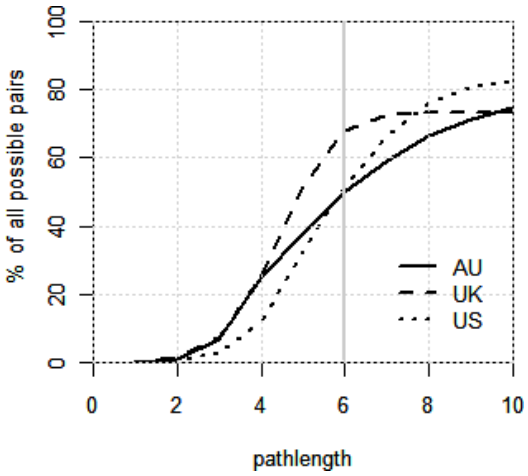


Figure 6.5: Reachability of the link graph ($\Gamma(FO, \tau_d)$). Cumulative percentage of pairs of pages (y-axis) that have a path between them of less than *pathlength* (x-axis). The value of six clicks is highlighted in the figure.

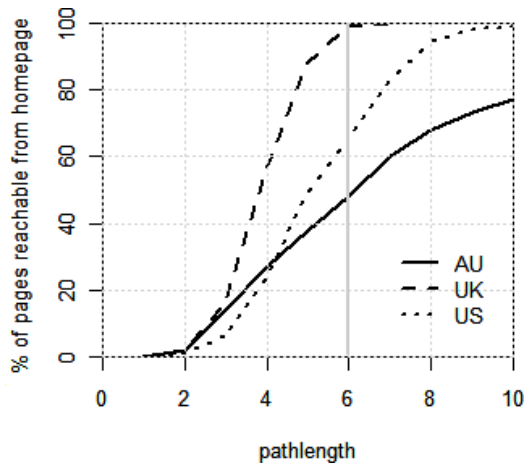


Figure 6.6: Reachability from website home page. Cumulative percentage of pages (y-axis) that are less than *pathlength* (x-axis) from the homepage. The value of six clicks is highlighted in the figure.

only about half of the content is accessible within six clicks on the other two websites.

As our crawl was started from the homepage, all pages in our sample are accessible and the lines in Figure 6.6 converge to 100%. However, different number of clicks is needed for each website to reach this 100%. Such number of clicks is related to, although smaller than the diameter of the website. The reachability plot gives a fuller picture of the website interconnectedness than

diameter as it shows the distribution of path lengths and can reveal if the large diameter is due to exceptional few pages or if a substantial part of the website is inaccessible.

Homepage reachability yields the following ranking: 1. UK: 68.17%, 2. US: 51.41%, and 3. AU: 49.9% (Figure 6.6). If we assume that a good foreign office website should make its content accessible within less than 6 clicks to users arriving at the homepage of the website, than the UK would be a much better website than both AU and the US. However this metric treats each page within the website equally and it needs to be verified if such metrics correlate with real user performance.

The structural properties that have an intuitive interpretation in terms of navigability provide an alternative view of the websites. In the next section we summarize the results of the user experiment and rankings by metrics and discuss how these two correspond.

6.4.3 Link structure of websites and user performance

In this section we compare the results of user experiment as measured by success-rate with the structural properties of the websites. We are in particular interested in possible correlations between the two different approaches to evaluate websites.

In treatment 1 there was no evidence for any difference in performance of users on the three websites. It is also likely that the performance of the websites will be influenced by their link structure to a smaller extent in this treatment given the machine learning approaches based on page features that are employed by search engines. The content of the pages may also override the effect of the link structure of the website as most search engines routinely discard the internal links for pagerank computation.

In treatments 2 and 3, most of the users navigated extensively and in treatment 3 they could not take any shortcuts. We further discuss the two treatments 2 and 3. Table 6.10 compares results of treatments 2 and 3 of the user experiment with the website metrics we reported in Section 6.4. Not all differences between countries are statistically significant. While we can say for sure that the UK is performing better than the US, Australia's performance is related to whether internal search is allowed or not.

Treatment 2 ranking matches the ordering by size. Treatment 3 ranking also matches the ordering by size, and in addition by diameter, reverse *LSCC* and $\Gamma(FO, home, 5)$ reachability.

In the table we see that the smaller the *LSCC* (in %), the better subjects perform in treatment 2. Possibly, too many connected pages confuse subjects, present a larger space of choices and make it hard to navigate. In such a case

	user experiment		link structure metrics			
Rank	success-rate		**size**	***LSCC* size (%)**	**average distance**	$\Gamma(home, 5)$ **(reachability)**
	decr		incr	decr	incr	decr
	T2	T3				
1	UK, AU	UK	UK	AU	UK	UK
2		AU, US	AU	US	US	US
3	US		US	UK	AU	AU

Table 6.10: A summary of results of the user experiment and website link structure metrics. Each column is ordered according to the rank of the respective country on that measure. The sorting order (decreasing/increasing)is specified above each column. This order has been chosen so that the websites with intuitively better characteristics are generally on the top and the ones with less advantageous properties on the bottom. Multiple countries in the same cell mean that there is no statistically significant difference between the two countries according to the respective measure.

it would be better to have a small part of the website strongly connected with the rest just leading out of it. However such a setup has the disadvantage for users to be stuck in the OUT component.

Ranking of websites by $\Gamma(FO, home, 5)$ reachability corresponds to results of treatment 3. For the UK about 70% of its content is accessible within that distance, while for both AU and US it is only about 50%. Intuitively, larger proportion of UK website is accessible to users with exploration depth of five. This could have an effect on user performance.

However, such match in rankings is not a proof of causal relationship and represents a connection in ranks only. In other words, the correlation does not reflect linear or higher order dependencies and is therefore of limited utility in constructing explanatory models. US for example is six times bigger than UK but does not perform six times worse.

Even though our study had over 100 participants, for the comparison of structural metrics and user performance on each website we effectively have a sample the size of three only. This limits our ability to extrapolate these results to other similar websites.

6.5 Summary

The main results of our, multi-faceted study can be divided in five parts: i) differences between the websites ii) differences between the treatments, iii) general observations, iv) structural properties of websites and v) the correlations between website structure and user performance.

Differences between the websites The three websites differ significantly in their performance. According to our experiment, UK website provides the information in the most accessible manner so that it can be much better reached when relying on navigation only than for any of the other two websites. Australia's website, however, is doing equally well to the UK when users are free to use the internal or external search engine, despite its poor performance in treatment 3 where users had to navigate. The US foreign office's suboptimal design is saved only by the external search engines, even the internal search engine does not seem to cope with the sheer size of the data presented on this website. These results indicate that the navigational structure of AU and US is suboptimal and that these two websites rely heavily on the search facilities – and in the case of US the essential search is, worryingly, provided by external third party.

Differences between the treatments We have seen that the user performance decreases with increasing restrictions from treatment 1 to treatment 3. Still some of the websites were affected more than other. While UK was hardly affected at all, US suffered in both treatments 2 and 3 when users could not take advantage of global search engines.

The use of the internal search facility is positively correlated with poor performance of users. Still, its availability actually increases users' performance. While this may seem paradoxical, it is due to users resorting to internal search when they are lost. The availability of internal search is beneficial for users and may help to compensate for other design problems.

The observation that users find the information easier when using global search engines than when navigating the website directly has interesting implications – campaigns for users to bookmark the website may be counterproductive and actually decrease user's ability to find information quickly and efficiently.

General observations Unrestricted treatment 1 provided us with an opportunity to make some general observations about general information seeking habits of users. We have seen high proliferation of Internet search use

with virtually all users using a global search engine at least once. Still subjects needed to navigate towards the final page with an average user clicking six times. The governments of the respective countries attracted between 73–83% of queries with the remaining 17–27% queries answered by websites out of their control.

Structural properties Although the three websites have equivalent tasks and mission, the website structure differs significantly. US is the largest website with 129K pages while UK and AU are roughly comparable at 24K and 33K pages respectively. The website diameters differ too and surprisingly do not correspond to the website size as the largest US(17) has smaller diameter than AU (38). The degenerate bow-tie structure of the websites are also very different, with AU having a largest strongly connected component of 89% compared to UK' 65%. Although it is small, our sample represents diverse ways of organization and presentation of equivalent content. The differences are likely to affect the user experience.

Structural properties and user performance Our study with 134 participants enabled us to draw statistically significant conclusions about the three representants of foreign office websites. We also compared the structural properties of the websites with their performance. However informative, these conclusions are not statistically significant as in these comparisons we need to treat the data as a sample of size three and any extrapolation to other FO websites is problematic. Still we observed that the bigger websites in our sample performed generally worse than leaner websites. It is surprising how big the foreign office websites are given the limited information they are expected to provide. Out of the structural metrics that have intuitive justification we cannot rule out the effects of diameter, reachability and the size of *LSCC* on the performance of navigating subjects who do not use search facilities. More studies are needed on a larger sample of websites to establish the statistical significance of website structural effects on the user performance.

Chapter 7

Exploiting homophily for better item recommendations

In this chapter we describe a way to exploit homophily to improve item recommendation. Results presented in this chapter have been reported in [178].

It is widely recognized that consumers suffer from information overload when using the Web. Recommender systems can help consumers by suggesting possible items of interest, based on an analysis of other users' preferences. Recommender systems utilize history of users' past preferences, e.g. items they purchased, movies they watched, ratings they provided, etc. When recommender systems work well, they can simultaneously benefit both the consumer and the retailer – a user's experience is improved, resulting in improved consumer loyalty and trust, and ultimately in increased revenues.

Recently, there has been very strong growth in social network applications. Social networking is at the core of sites such as MySpace, LinkedIn, Facebook, Yahoo!360, Friendster and many others[1]. Social networking features are increasingly being adopted by other sites for which social networking was *not* the primary focus, e.g. MyWeb, Flickr and Delicious. Data from social networks provides insight into the structure of communities, the spread of information, and may identify thought leaders, innovators, hubs and authorities and other influential participants.

While social network data is interesting in itself, there exists an opportunity to leverage this information by combining it with preference information to provide better recommendations.

In Section 4.2 we present an overview of related work. Despite the amount of work on recommender systems, which we have reviewed in Section 4.3,

[1]For an extensive list of social networking sites see http://en.wikipedia.org/wiki/List_of_social_networking_websites

there has been little published on combining recommender systems and social networks to improve recommendations. Using social networks for recommendations has been previously suggested [231, 23]. However, the benefits have not been quantified or demonstrated, nor have any algorithms been evaluated.

In this chapter we look at several ways of using social network knowledge to improve recommendations. We describe new social recommenders (SOC) and combined, socially-informed collaborative filtering (SCF) algorithms. SCF algorithms combine knowledge of a user's social network with traditional collaborative filtering to improve recommendations. We perform empirical comparison of these new algorithms with pure collaborative filtering algorithms on real world datasets from the social photo sharing site Flickr and from the social networking site Yahoo!360. We use weak and strong generalization test procedures to compare the performance of combined algorithms to pure collaborative filtering recommenders. We also use a new-user test to show that the benefit of using social information increases with increasing sparsity of the user preference information.

Section 7.1 first describes the new algorithms. Many of the algorithms compared have free parameters that must be chosen. Section 7.2 describes the two dataset used for the algorithm evaluation. Section 7.3 describes how these parameters were assigned and describes the weak, strong and new-user generalization tests that were used to compare the various algorithms. The experimental results are discussed in Section 7.4 and a summary and discussion are provided in Section 7.5.

The algorithms we evaluate in this study can be categorized into four classes; (1) non-personalized (baseline), (2) pure collaborative filtering, (3) pure social-based and (4) socially-informed collaborative filtering that combines the social knowledge of (3) with the collaborative filtering of (2). We already introduced classes (1) and (2) in Chapter 3. We are going to discuss each of the classes (3) and (4) next.

7.1 Socially informed algorithms

The network of social contacts is in fact an implicit declaration of similarity between the users connected. This information is complementary to the information mined from preference history. We describe algorithms that take advantage of this social information. First we describe an algorithm based purely on the social information and then we describe a fusion of this social recommender with collaborative filtering based recommenders.

The reader is reminded that a recommender consists of two parts *score*

calculation and *recommendation generation* (Chapter 3), the latter being the same for all recommenders.

7.1.1 Social-based algorithm (SOC)

Social algorithm SOC is based on homophily of the social network. Users connected by social ties are more likely to be similar than a random pair of users. It seems therefore promising to recommend items based on popularity between social contacts rather than global popularity as algorithm POP does. We implemented a purely social-based algorithm, denoted SOC, which makes recommendations based on the social popularity of items. For each user, u, we identify their k nearest neighbors, K_u^k, according to distance in the social network, $d(u, v)$ (Section 2.2). The distance corresponds to the number of hops from user u to user v in the directed social network graph. Directly connected users have distance 1. The ranking of nodes with the same distance is randomly determined.

The SOC algorithm effectively performs a breadth-first crawl, starting from the active user and aggregating the preferences of users encountered. We considered two different weighting methods, denoted 'const' and 'distance', for the aggregation of neighbor preferences. For a method, t, the scores for individual items are determined by:

$$w_{ui}^t = \sum_{v \in K_u^k} f_{vi} \cdot \theta_{uv}^t \tag{7.1}$$

where

$$\theta_{uv}^t = \begin{cases} 1 & \text{if } t = \text{`}const\text{'} \\ \frac{1}{d(u,v)} & \text{if } t = \text{`}distance\text{'} \end{cases} \tag{7.2}$$

The first method, $t =$'const', corresponds to the case where the score of an item, i, is only based on the number of neighbors of u who have this item in their favorite set.

The second method, $t =$'distance', is similar, but is inversely weighted by each neighbor's distance from u, i.e. closer neighbors are given more weight. This reflects the assumption that users closer to each other are more similar.

The two social-based algorithms are denoted for example SOC(200,'const'), which corresponds to a nearest neighborhood of $k = 200$ and constant weighting.

Algorithm 5 presents pseudocode of the score computation step of SOC recommender. FIFO represents a first-in-first-out queue. The operation FIFO.append(a) places user v at the end of the queue and operation FIFO.get() removes and returns the user who is at the front of the queue.

Algorithm 5 SOC(a, k, t)

```
1: c ← 0
2: FIFO.append(a)
3: while u ← FIFO.get() do
4:    for all v : e_uv = 1 do
5:       FIFO.append(v)
6:    end for
7:    for all i : F_u do
8:       case t
9:        'const':
10:           w_a i ← w_a i + f_ui
11:       'distance':
12:           w_a i ← w_a i + f_ui / d(a,u)
13:      end case
14:      c++
15:      if c > k then
16:         return w_a
17:      end if
18:   end for
19: end while
```

SOC algorithm combines preferences of other users – in that it is similar to UU algorithm. It has an advantage over UU in that it only needs to inspect k social contacts of the active user compared to computation of similarities between active user and *all* other users in the system. SOC is therefore more scalable than UU.

7.1.2 Social collaborative filtering (SCF)

Pure collaborative filtering and pure social-based algorithms provide recommendations based on complementary information. It therefore seems natural to combine both in a single algorithm to further improve recommendations. We propose the following algorithm to integrate both sets of information.

For a given collaborative filtering algorithm, CF, and a given social algorithm, SOC, we first calculate the corresponding item scores for each user, w_{ui}^{cf} and w_{ui}^{soc}, as described earlier. Then, for each user, these two score vectors are normalized such that $\sum_i w_{ui}^{cf} = 1$ and $\sum_i w_{ui}^{soc} = 1$. The two normalized score vectors are then combined using one of the following methods to calculate the final item scores:

- Probability-like: $w_{ui} = 1 - (1 - w_{ui}^{cf}) \cdot (1 - w_{ui}^{soc})$

- Weighted sum: $w_{ui} = \alpha w_{ui}^{cf} + (1-\alpha) w_{ui}^{soc}$
- Mean: $w_{ui} = \frac{1}{2}(w_{ui}^{cf} + w_{ui}^{soc})$
- Maximum: $w_{ui} = max(w_{ui}^{cf}, w_{ui}^{soc})$
- Minimum: $w_{ui} = min(w_{ui}^{cf}, w_{ui}^{soc})$

Finally, once again, items are sorted in decreasing order of score, excluding items in the user u's favorites, and the top-N items are recommended. Algorithm 6 presents pseudocode of score calculation step for SCF algorithm. Note the normalization on lines 4 and 5 preceding the application of combination rule.

Algorithm 6 SCF(a, cf, soc, v, α)

1: **for** each neighbor u **do**
2: $w_a^{CF} \leftarrow cf(a)$
3: $w_a^{SOC} \leftarrow soc(a)$
4: $w_a^{CF} \leftarrow \frac{w_a^{CF}}{\|w_a^{CF}\|_1}$
5: $w_a^{SOC} \leftarrow \frac{w_a^{SOC}}{\|w_a^{SOC}\|_1}$
6: **case** v
7: 'prob':
8: $w \leftarrow 1 - \big((1 - w_a^{CF}) \cdot (1 - w_a^{SOC})\big)$
9: 'wsum':
10: $w \leftarrow \alpha w_a^{CF} + (1-\alpha) w_a^{SOC}$
11: 'min':
12: $w \leftarrow min(w_a^{CF}, w_a^{SOC})$
13: 'max':
14: $w \leftarrow max(w_a^{CF}, w_a^{SOC})$
15: **end case**
16: **end for**
17: **return** w

SCF algorithm is naturally slower than each of the algorithms it is composed of. The combination step is very quick though, thanks to the simplicity of combination rules used.

7.2 Datasets

In our study, we used two datasets: Yahoo! 360[2] dataset and Flickr[3] dataset. Both sites provide social networking functionality and also let users express their preferences. However, they differ substantially in their main purpose: Yahoo!360 is primarily a networking site while Flickr is a photo-sharing site. We now describe these datasets in detail.

Table 7.1 provides a summary of the Yahoo!360 and Flickr data. We report the two thresholds τ_c and τ_i that were used to preprocess the datasets, corresponding to the minimum number of contacts and minimum number of favorites respectively. Note the difference in acquisition methods; Yahoo!360 has been crawled by a focused crawl of the contact graph. while Flickr has been obtained from the complete dump of the Flickr database.

Dataset	Yahoo!360	Flickr
τ_c	20	100
τ_i	20	100
users $\|\mathcal{V}\|$	2191	820
items $\|\mathcal{I}\|$	32,922	355,389
favorites $\|\|F\|\|_1$	85,111	583,932
ratings density	1.2×10^{-3}	2.0×10^{-3}
social connections $\|\|E\|\|_1$	38,870	85,042
social network density	8.1×10^{-3}	1.2×10^{-1}

Table 7.1: Yahoo!360 and Flickr datasets

7.2.1 Yahoo!360 dataset

Yahoo! 360 is a social networking site where users can create their own profile, link to other users, and publish their hobbies, interests, photos, blog and other information. Users interact on the site by sending messages, leaving comments and testimonials for their friends and joining discussion groups. Users are encouraged to create a list of their favorite music, movies, books, TV shows and their interests. This list of favorite items is used by us to create a user-item matrix. This list contains items in no particular order that have been entered as comma separated text. There is a separate text box for each item category.

[2] http://360.yahoo.com
[3] http://flickr.com

Data was collected between November 21 2006 and January 10 2007. We started from an arbitrary user and recursively performed a focused crawl of all their contacts. We followed the Algorithm 1 described in Section 2.6. At each step we downloaded a new contact with the highest indegree[4]. This permitted an efficient crawl of a dense subset of the social network.

We collected data in the form of network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and attribute data $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$ where $\mathcal{V}$ is the set of users, $\mathcal{E}$ are edges or social contacts between users, $\mathcal{I}$ is a set of all items and $\mathcal{F}$ stores user-item co-occurrences signifying which user has listed an item between his favorites.

For each user, we parsed their list of favorites and cleaned item names by converting them to lowercase and removing all whitespace and punctuation. In total we collected 789,523 user–item pairs (favorites) from 97,606 unique users. There were 235,762 unique items. We treat favorite books, movies and TV shows equally and refer to all of them as items. Then, we removed items with extremely long descriptions[5] and selected a subset of active users – first we selected users who had at least 20 favorite items and then retained those users with at least 20 contacts. After this filtering we obtained a dataset with 2191 users, 32,922 items, 85,111 favorites and 38,870 social connections. The resulting user-item matrix density was 0.1% and contact matrix density 0.8%. Note that not all users have more than 20 contacts in the resulting dataset as some of their contacts were originally to users who had less than 20 items or 20 contacts themselves and were therefore removed. This filtering results in a more dense social network than random sampling. We also refer to this dataset as Y!360.

7.2.2 Flickr dataset

Flickr is a photo sharing site that allows users to publish photos, tag them, comment on them, mark favorites and keep a list of their friends and contacts. The Flickr dataset we used was a subset of a snapshot of the Flickr database from June 2005. The full snapshot contained 2,618,702 social network connections with 434,812 users having at least one contact and 420,497 users being a contact of at least one other person. These connections were categorized as friend, family and other. For simplicity we did not distinguish between different connection types. The Flickr snapshot also contained explicit user preference information such as favorite pictures a user has identified or "bookmarked". The dataset contains 2,282,529 user–favorite picture pairs

[4]The indegree (number of inlinks) is dynamically computed based on the crawl so far.

[5]Some users enter things like "I would really have to think about this a little bit more" when asked for their favorite movies

(favorites) with 60,904 unique users that have at least one favorite picture each and 1,077,782 unique pictures that are favorite of at least one user.

Again the data is represented, in a form consistent with Section 2.2, as network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and attribute data $\mathcal{A}(\mathcal{V}, \mathcal{I}, \mathcal{F})$. $\mathcal{V}$ is the set of users, $\mathcal{E}$ are edges or social contacts between users, $\mathcal{I}$ is a set of all pictures (items) and $\mathcal{F}$ stores user-picture co-occurences signifying which user has listed a picture between his favorites.

From the complete Flickr snapshot, we selected active users with at least 100 contacts and 100 favorites. The filtered dataset contains 1,357,143 favorites from 4,142 users on 680,857 items and 377,185 social connections. We refer to this dataset as Flickr.

7.2.3 Social network homophily

For social recommendation to work, users connected by social ties need to be on average more similar than two random users.

The homophily coefficient, $\chi(i)$, reflects the degree to which users who like item i tend to associate with other users who also like item i. We demonstrate the use of this coefficient on dataset Yahoo!360. We show in Table 7.2 the most and least homophilous items in the dataset. We can see that more rare items tend to induce higher homophily. This observation is in agreement with results published by Adamic [5]. Only coefficients based on 100 and more i-homophilous links are included. All $\chi(i)$ coefficients are bigger than 1 which means that all the property induce positive homophily.

The generally high homophily in our dataset suggests that item popularity in the social neighborhood may be a better predictor for recommendations than global popularity, provided there is enough social data for the active user.

7.3 Test procedure

We use weak and strong generalization tests [141] as well as a new-user test to compare the algorithms described previously. We first introduce the three individual tests separately and then we describe the way we have split the datasets to be able to perform all of them.

7.3.1 Weak generalization test

Weak generalization tests performance of algorithm in terms of prediction of active user's favorite items not seen during training. For this, the dataset

$\chi(i)$	item i	i-homophilous edges
149.67	bleach	120
99.22	naruto	975
95.60	fullmetalalchemist	158
86.70	inuyasha	553
73.17	manga	135
54.94	meditation	146
35.87	godsmack	104
34.84	neosoul	134
34.00	cricket	150
...		
4.54	photography	1547
4.44	friends	1378
4.31	travel	1454
4.24	cooking	1231
3.80	csi	1867
3.46	movies	1709
3.05	reading	3744
2.90	music	3529

Table 7.2: Associativeness coefficient $\chi(i)$ that quantifies how much people with interest i in Yahoo!360 dataset tend to flock together. The most homophilous tend to be the more rare hobbies while the popular interests exhibit lower homophily.

is split in two parts. One part is used for training the algorithm and the other consists of withheld items and is used for testing. Algorithm is asked to provide recommendations for each user encountered during training and the items recommended are compared to items withheld for the user. The division of the dataset is indicated in Figure 7.1.

7.3.2 Strong generalization test

Strong generalization tests the performance of algorithm for previously unseen users. First the users are split into two groups – "weak users" and strong users". For each strong user items are divided into revealed items and items withheld for testing. Algorithm is trained on data from weak users first and then asked to provide recommendations for each "strong user" based on the items revealed for her. The recommendation is then compared to items that have been withheld previously. The division of the dataset is indicated in Figure 7.2.

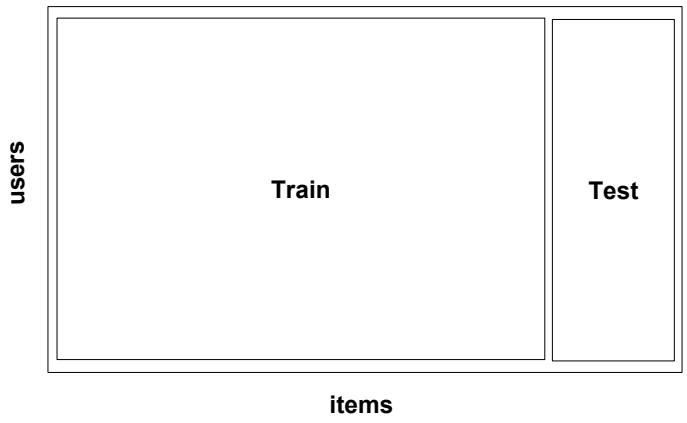


Figure 7.1: Weak generalization test.

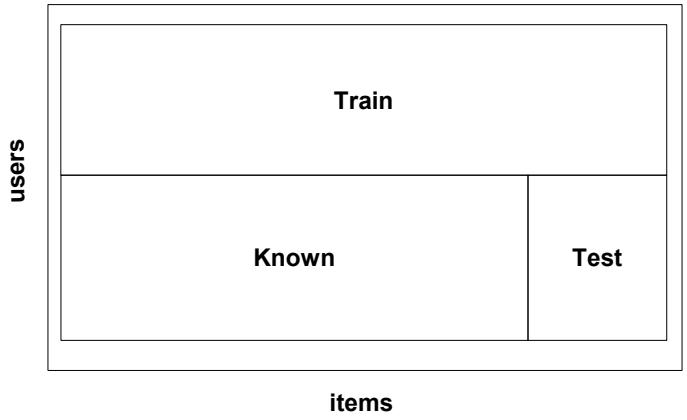


Figure 7.2: Strong generalization test.

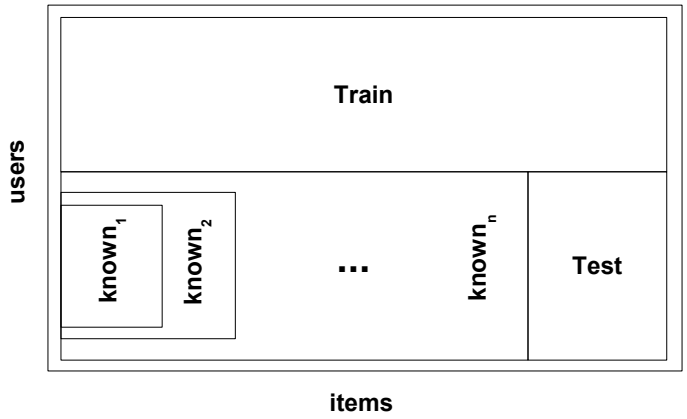


Figure 7.3: New-user generalization test.

7.3.3 New-user test

For new-user testing, we wish to test the predictive capabilities of the various algorithms when we have little or no history of the user. The reader is reminded that it is our hypothesis that the use of complementary social information may strongly improve performance for this class of users. New-user test simulates new users arriving in the system and providing their preferences gradually. For this purpose the users are first divided, similarly to strong generalization test, into "weak users" and "strong users". The division of the dataset for new-user test is indicated in Figure 7.3. To simulate new users, the items in the known set of the strong user group are partitioned into 10 subsets, $known_1$, through $known_{10}$, where $known_1$ is a subset of $known_2$, $known_2$ is a subset of $known_3$, etc. Thus, $known_1$ contains the fewest items for each user, and $known_{10}$ the most. As with the strong generalization test, data in the $known_i$ subset represents the history of the user, and is used to make recommendations. Recommendations are as usually compared to the strong test set withheld from data of strong users.

7.3.4 Datasets split

We now describe the partitioning of the dataset for training and test purposes. To perform the weak generalization, strong generalization and new-user tests we split the dataset as indicated in Figure 7.4. For a given dataset, Yahoo!360 or Flickr, we randomly partition the users into two equal sized

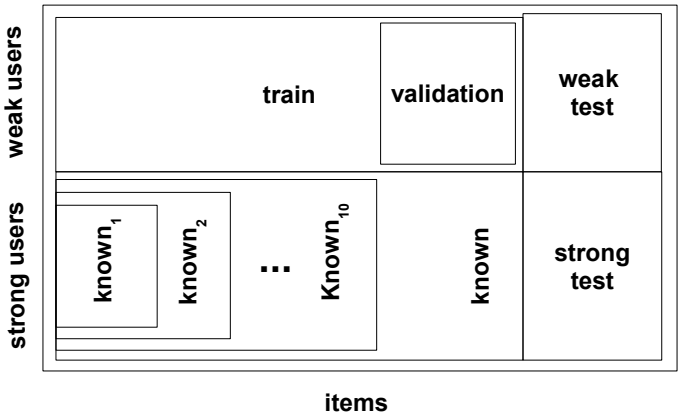


Figure 7.4: Dataset split for generalization tests.

groups – "weak users" and "strong users".

For the weak generalization test we further partition the data of weak users into a test set and a training set. The test set contains n_{wtest} items from each user, i.e. a subset of the items that each user is known to have consumed. The remaining items for each user form the training set. From the training set, we further withhold another subset of items from each user, this set being used for validation. After training, we test each algorithm's predictive capability based on the test set derived from the weak users. That is, the same users are used for training and testing, although, of course, the training and tests sets are disjoint.

For strong generalization, testing is performed using the strong user group, no part of which has been seen during training. For the strong user group, we randomly select n_{stest} items from each user, which we withhold to test the predictions of each algorithm. The remaining items for each user form a known or "observed" set, that is used to compute the recommendation. The items in the known set simply represent the items that each user has consumed so far and which are used by algorithms item-item and user-user to make recommendations.

For weak generalization, we train on a set of users for which we have withheld items that they have previously consumed. After training, we test whether the algorithm predicts these withheld items. For strong generalization, we use the *same* training set as for weak generalization. However, after training, we test whether the algorithm predicts items for unseen users,

i.e. these users were not part of the training set. Finally, the new-user test, takes users from the strong generalization test set, but only reveals a subset $known_i$ of their consumed items, in order to model the case where we have little prior knowledge of a user. All training, whether for weak, strong or new-user generalization, is performed on the same training data.

7.3.5 Parameter selection

Before training each algorithm on the training set, we perform parameter selection using a validation set of n_{val} items per user which we extract from the training set to reduce overfitting. For UU and II algorithms we need to learn the number of nearest neighbors, k, and minimum overlap, τ. For SOC algorithm we need to learn the number of neighbors k, and choose a weighting type of 'const' or 'distance'. Finally, we need to learn the α parameter for the SCF algorithm using the 'wsum' combination rule. During training UU(70,1), II(100,2) and SOC(200,'distance') performed best on the Yahoo!360 validation dataset. On the Flickr dataset UU(70,1), II(100,2) and SOC(60,'distance') performed best. For the combined algorithm, SCF, the optimal values of parameter α were $\alpha_{UU} = 0.3$ and $\alpha_{II} = 0.05$ on Yahoo!360 and $\alpha_{UU} = 0.01$ $\alpha_{II} = 0.2$ on Flickr. Parameter selection for algorithms SOC, and SCF is detailed in Appendix B.1 and we use the same naming convention for them as for the collaborative filtering recommenders. These parameter values were used in the experimental results presented shortly.

7.3.6 Metric

Several metrics can be used to measure the performance of recommender systems. We chose the *F1* metric which combines traditional precision and recall into a single metric. Other measure include hitrate, reciprocal hitrate, precision, and recall. Although not reported here, we considered all these measures, each of which produced similar results. For precision and recall, we assume that the set of withheld items is relevant. Thus,

$$\text{prec}_N = \frac{1}{n} \sum_u \frac{h}{N} \tag{7.3}$$

$$\tag{7.4}$$

where n is the number of users tested, h is number of recommended items matching one of the items withheld, $test(u)$ set of withheld items for user, u, and N is the number of recommended items.

The metric $F1_N$ gives the same weight to prec_N and recall_N and combines them in a single metric, defined as

$$F1_N = \frac{1}{n} \sum_u \frac{2 \cdot \text{prec}_N \cdot \text{recall}_N}{\text{prec}_N + \text{recall}_N} \tag{7.5}$$

F1 metric has been introduced in information retrieval research in the context of labeled datasets, where each document is labeled as relevant or not relevant to a particular query. The set of documents labeled relevant represents all documents that are relevant. Important difference in the way F1 metric (and precision and recall) are defined using just a subset of all possibly relevant documents. The preference data contains only a relatively small number of items for each user in the form of their purchase history or list of favorite items. There are many other items that the user would like but which the user did not see or did not mark as favorite. This is particularly obvious in the case of user generated content such as pictures on photo sharing websites.

The calculation of F1 values over a limited set of relevant items results in lower absolute value of this metric because not all relevant items are recognized as such. Past evaluations of recommender systems reported lower values of F1 metric than classic information retrieval experiments. The absolute F1 values get smaller with the increasing size of the datasets and the increasing number of items in the datasets. This is demonstrated on the example of a simple random recommender in Appendix B.2.

7.4 Results

In this section we compare the performance of the various algorithms for weak generalization, strong generalization and new-user cold start, using the datasets Yahoo!360 and Flickr. All experiments were performed with the number of recommended items $N = 100$. This corresponds to a 10x10 grid of thumbnails for Flickr or a textual list of 100 favorite items for Yahoo!360.

The number of items withheld per user for weak generalization, strong generalization and validation were $n_{stest}^{Y!360} = n_{wtest}^{Y!360} = n_{val}^{Y!360} = 2$ and $n_{stest}^{Flickr} = n_{wtest}^{Flickr} = n_{val}^{Flickr} = 10$. The Yahoo!360 validation, weak test and strong test set contained 2192, 2192 and 2190 withheld user-item pairs. For Flickr, there are 4100 withheld user-item pairs.

Yahoo!360			
weak		strong	
alg	F1	alg	F1
SCF(prob,II)	0.00889	SCF(prob,II)	0.00885
SCF(max,UU)	0.00869	SCF(prob,UU)	0.00874
SCF(prob,UU)	0.00861	SCF(max,II)	0.00856
UU	0.00843	SCF(wsum,II)	0.00849
SCF(wsum,UU)	0.00841	II	0.00849
SCF(max,II)	0.00827	SCF(max,UU)	0.00827
SCF(wsum,II)	0.00827	SCF(mean,II)	0.00827
SCF(mean,UU)	0.00823	SCF(mean,UU)	0.00825
II	0.00821	SCF(wsum,UU)	0.00825
SCF(mean,II)	0.00798	UU	0.00802
SOC	0.00755	SOC	0.00722
SCF(min,UU)	0.00689	SCF(min,UU)	0.00704
SCF(min,II)	0.00619	SCF(min,II)	0.00627
POP	0.00567	POP	0.00550

Table 7.3: Weak and strong generalization test on Yahoo!360 using the F1 metric. Algorithms are ranked in order of performance.

Algorithms comparison

The algorithms compared include POP, UU, II, SOC, and SCF. We compare several variants of SCF based on either II or UU and using one of voting rules 'min', 'max', 'prob', or 'wsum'.

A number of observations can be drawn from Tables 7.3 and 7.4. First, for both datasets, we observe that performance in the strong generalization test is, as expected, generally slightly worse than in weak generalization test. All tested algorithms outperformed the global POP algorithm (except SCF(min,II) on Flickr). Interestingly, the superior performance of the SOC algorithm over POP is indicative of the strong similarity between users connected in a social network. Nevertheless, the SOC algorithm lags behind traditional collaborative filtering algorithms, UU and II.

We looked at a number of combined social-collaborative filtering recommenders (SCF), that differed only in the manner in which the information from the two sources, CF and SOC, were combined[6]. Combining item-item

[6] We note that for the case of combining information from different classifiers, Duin and Tax [61] have observed that different combination rules result in different performance and that these differences are data dependent. Kittler et al. [119] discussed classifier fusion using a variety of combination rules in two scenarios i) classifiers with distinct

Flickr			
weak		strong	
alg	F1	alg	F1
SCF(wsum,UU)	0.00381	SCF(prob,UU)	0.00386
UU	0.00381	SCF(mean,UU)	0.00377
SCF(mean,UU)	0.00368	SCF(wsum,UU)	0.00373
SCF(prob,UU)	0.00368	UU	0.00373
SCF(max,UU)	0.00364	SCF(max,UU)	0.00373
SCF(prob,II)	0.00341	SCF(min,UU)	0.00328
SCF(max,II)	0.00341	SOC	0.00293
II	0.00337	SCF(prob,II)	0.00266
SCF(wsum,II)	0.00328	SCF(wsum,II)	0.00262
SCF(min,UU)	0.00297	II	0.00262
SCF(mean,II)	0.00297	SCF(max,II)	0.00262
SOC	0.00275	SCF(mean,II)	0.00239
POP	0.00160	POP	0.00182
SCF(min,II)	0.00142	SCF(min,II)	0.00151

Table 7.4: Weak and strong generalization test on Flickr using the F1 metric. Algorithms are ranked in order of performance.

collaborative filtering with SOC using "probability-like" fusion, results in the best performing algorithm for both weak and strong generalization. For the Yahoo!360 dataset and weak generalization, the SCF(prob,II) algorithm improves performance by over 8% compared with the item-item (II) algorithm alone. User-user (UU) collaborative filtering outperforms II, but the SCF(prob,II) algorithm still performs 5% better that the user-user algorithm. For the Yahoo!360 dataset and strong generalization, item-item CF outperform user-user CF, and our SCF(prob,II) is 4% better than II alone and 10% better than UU.

For the Flickr dataset, the improvements are smaller. For weak generalization, we do not observe any improvement over user-user CF alone. However, for strong generalization, we observe that the SCF(prob,UU) outperforms UU collaborative filtering by 3%. We hypothesize that the reduced improvement is due to the fact that the number of known favorites for each user is much greater in the Flickr dataset (90) compared to the Yahoo!360 dataset (18). Thus, the additional information provided by the social network

representations – different features and ii) classifiers with the same representation/features. Shows that fusion is in fact a multistage classification where the probabilities from the two independent classifiers serve as features for the combining classifier.

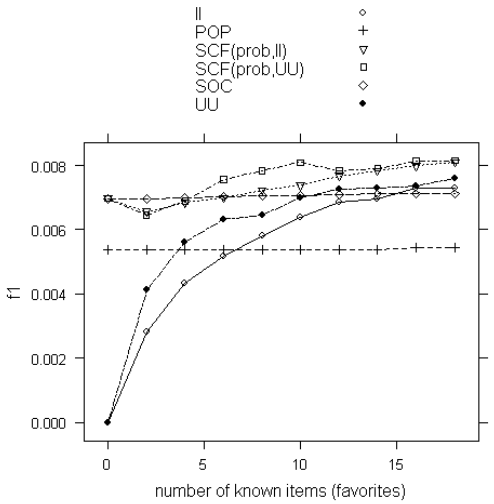


Figure 7.5: New-user cold start for Yahoo!360 dataset. The horizontal axis denotes the number of favorite items observed.

is less useful in this case.

The new-user test simulates a cold start situation by varying the number of known favorites (items), n, provided by active users. Experimental results are provided for the Yahoo!360 dataset The POP algorithm serves as a baseline and is seen to perform worse, except for very limited information, i.e. for $n = 0, 2, 4$.

The item-item (II) and user-user (UU) recommenders have extremely poor performance for $n < 12$. In contrast, the combined social-collaborative filtering recommenders perform well even for very limited user information, i.e. $n = 2$. In the extreme case of $n = 0$, the pure social algorithm SOC and both SCF(prob,II) and SCF(prob,UU) (combining SOC with CF using "probabilistic-like" fusion) still manage to provide reasonable quality recommendations while CF algorithms fail. For the case of $n = 2$, the SCF algorithms provide an improvement of over 100% compared with item-item alone and over 50% compared with user-user CF. As expected, the performance gain of SCF over II and UU diminishes as users provide more ratings, i.e. the number of know favorites increases. However, in the range observed, the performance of the SCF algorithm exceeds that of II or UU alone.

Finally, for very limited user rating information, e.g. $n < 4$, the use of a purely social-based algorithm (SOC) may be preferred.

7.5 Summary

We demonstrated how knowledge of social network can be used to improve recommendations to users. This additional information is most useful for the new-user cold start situation, where, on the Yahoo!360 dataset, we demonstrated performance improvements of over 100% over traditional item-item and 50% over user-user collaborative filtering. Even when there was more rating available for users, the combined algorithm outperformed the traditional methods. For the Yahoo!360 dataset and weak generalization, improvements of 5% and 8% were observed for SCF over user-user and item-item algorithm, and 10% and 4% for strong generalization. For the Flickr dataset, no improvement was observed for weak generalization, but for strong generalization, we observed a 3% improvement over user-user algorithm alone. This variation across datasets may be due to the fact that the Flickr dataset provides substantially more known favorites (items) for each user (90) compared to that available in the Yahoo!360 dataset (18). Thus, the additional information provided by the social network is relatively smaller for the Flickr case. Nevertheless, for strong generalization, the incorporation of social network information always improved performance.

Even pure social recommenders performed surprisingly well – much better than global popularity (POP) and outperform CF algorithms when little data from the active user is available. Experiments on the Yahoo!360 dataset showed performance improvements of 100% and 50% over item-item and user-user collaborative filtering when the number of known favorites for a user was very small, i.e. only two. As expected, as more information about a user becomes available, the relative improvement becomes smaller, but performance of the combined algorithm was always better. We also note that in the case of a completely new user for which no item (favorites) information is available, traditional collaborative filtering fails. However, if social network information is available, good quality predictions are still possible.

Improving recommendations has real benefits as it leads to higher user loyalty, improved value of online services to users, and, consequently, often leads to improved revenue. However, the performance observed by new-users may be much worse than for regular users due to the sparsity of information initially available about new-users. There is therefore a risk that new-users will not remain with the service. However, if complementary social networking data is available, then we demonstrated that significant improvements in performance can be achieved.

We expect social network data to be increasingly available in the future as networking features are integrated into more sites and become easier to use. Sites that combine the social data with traditional CF data may well

have a commercial advantage.

The SOC algorithm also has a computational advantage over the traditional user-user collaborative filter, which needs to compute similarity between the active user and all other users. This usually cannot be precomputed. In contrast, the SOC algorithm inspects only social contacts and does not need to dynamically compute the similarity. Moreover, the number of social contacts needed to be inspected is of the order of hundreds, which is much less than the number of users that must be inspected for user-user CF algorithm.

Chapter 8

Discussion

In this chapter we restate the contributions of the work presented in this thesis and discuss possible future research directions.

8.1 Contributions

The goal was to analyze different networked environments and to study ways to improve performance of systems using the knowledge of these environments. We report contributions in three areas:

1. Citation network analysis
2. Website link structure analysis
3. Recommender systems and social networks

We compared two popular computer science citation databases CiteSeer and DBLP in terms of citation distribution and acquisition methods. For the citation distribution we showed how they differ and that Computer Science citation distribution is significantly steeper than citation distribution reported for Physics community, resulting in higher proportion of citation being accumulated by highly cited computer science papers. We also showed that the difference in acquisition methods between citation databases results in a bias against single author papers and presented probabilistic models of the acquisition which result in a similar bias. The differences observed serve as a warning when these databases are used to compare scientific output and make decisions about funding.

We analyzed the internal link structure of three foreign office websites. We performed a navigational user study with 134 participants in a controlled

environment of computer lab. Our study showed significant differences between the websites in terms of navigability. We also point out possible effects of website size and structure on the performance of users when looking for information on these sites. Our experiment resulted, among others, in some common-sense recommendations to, especially e-government, website designers. Given the limited amount of information that users seek on a foreign office website and which the foreign offices are supposed to provide, optimal websites should be as lean as possible and the presentation of information should be supported by link structure that allows users to reach other relevant pages with the least effort.

We described novel algorithms, using the social network of users, that provide higher quality recommendations especially in the new-user cold-start situation. We performed an empirical quantitative evaluation of these algorithms on two real datasets from Flickr photo sharing website and Yahoo!360 social networking website. The benchmarks showed that our algorithms compare favorably with the state-of-the-art collaborative filtering methods and are significantly superior in a new-user cold start situation. These new algorithms may improve user experience, increase loyalty and also improve revenues of service providers.

We have shown in three different areas how network structures differ and how they influence the performance of systems. We also demonstrated how the knowledge of network structures may be used to improve the performance of these systems. More needs to be done and this work presents only a start of our future research.

8.2 Future work

We are working on a study of geographical citation networks using metadata automatically extracted from the CiteSeer dataset. We plan to investigate the questions "Which types of collaborations are more successful?" and "What are the geographical citation patterns?". Even though citation analysis is not a new area, citation patterns are changing in time and depending on geographical location.

In the website navigation area more studies are needed on other foreign office websites to establish statistically significant links between website structural properties and website usability. Also studies on other types of websites can bring useful insight into which of our observations are general and which are specific to foreign offices.

Our recommender systems algorithms present a first foray into the area of using social information to improve recommender systems. More sophisti-

cated algorithms based on support vector machines [48] for example may be more robust and be bale to exploit the full potential of social information. We also plan to apply our algorithms and new variants to other datasets such as the tag datasets of MyWeb[1] or ZoneTag[2].

[1] http://myweb.yahoo.com
[2] http://zonetag.research.yahoo.com

Appendix A

Website navigation analysis

A.1 Questions

The questions were the same for every treatment and for all countries. They had to be answered by selecting the correct option from a provided set of answers. In order to limit misunderstanding the questions were always explicitly referring to the country in question. The following is a list of our 10 questions we used in this analysis (here for the subjects in the Australian group). Questions for other countries differ only in the name of country used.

1. You want to travel to Vietnam as a tourist for two weeks. As an Australian citizen, do you require a visa to do so?

2. What is the address of the Australian embassy in Berlin/Germany? Please state the house number!

3. Official Australian documents that are going to be used abroad often need to be authenticated by an official Australian institution, to indicate that the document is not a fake. Does the Australian Department of Foreign Affairs and Trade authenticate documents?

4. You want to go to China for three weeks. Recently there have been reports on cases of avian flu / bird flu. Does the government of Australia advise its citizens against travel to China because of avian flu?

5. What is the opinion of the Australian government concerning: Is it safe for its citizens to travel to Ivory Coast/Cote d'Ivoire?

6. What is the Internet address of the French embassy in Australia?

7. As an Australian citizen: what should you do if your passport got stolen whilst you are abroad?

8. As an Australian citizen: In case you are arrested and imprisoned in a foreign country - will an Australian official (i.e. consul) visit you if you wish so?

9. What is the annual salary for Graduate Trainees starting to work for the Australian Department for Trade and Foreign Affairs?

10. What is the first name of the Australian ambassador in Israel?

A.2 Group differences

```
TukeyHSD(aov(SUCCESS~COUNTRY:TREATMEN),data=jointexp)
Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = SUCCESS ~ COUNTRY:TREATMEN)

$`COUNTRY:TREATMEN`
diff         lwr          upr     p adj
UK:1-AU:1  0.037253528 -0.08473416  0.159241211 0.9883705
US:1-AU:1  0.044410350 -0.07757733  0.166398033 0.9651767
AU:2-AU:1  0.017423675 -0.09937064  0.134217989 0.9999323
UK:2-AU:1  0.010022559 -0.10536624  0.125411356 0.9999990
US:2-AU:1 -0.107375262 -0.22416958  0.009419053 0.0980736
AU:3-AU:1 -0.099637410 -0.23225200  0.032977184 0.3087223
UK:3-AU:1  0.056657799 -0.07595679  0.189272392 0.9142545
US:3-AU:1 -0.141226966 -0.27061444 -0.011839489 0.0213800
US:1-UK:1  0.007156822 -0.11483086  0.129144505 1.0000000
AU:2-UK:1 -0.019829854 -0.13662417  0.096964461 0.9998198
UK:2-UK:1 -0.027230969 -0.14261977  0.088157828 0.9979933
US:2-UK:1 -0.144628790 -0.26142310 -0.027834475 0.0046262
AU:3-UK:1 -0.136890938 -0.26950553 -0.004276345 0.0374318
UK:3-UK:1  0.019404271 -0.11321032  0.152018864 0.9999416
US:3-UK:1 -0.178480494 -0.30786797 -0.049093017 0.0008971
AU:2-US:1 -0.026986676 -0.14378099  0.089807639 0.9982727
UK:2-US:1 -0.034387791 -0.14977659  0.081001006 0.9900883
US:2-US:1 -0.151785612 -0.26857993 -0.034991297 0.0023123
AU:3-US:1 -0.144047760 -0.27666235 -0.011433167 0.0225084
```

```
UK:3-US:1  0.012247449 -0.12036714  0.144862042 0.9999984
US:3-US:1 -0.185637316 -0.31502479 -0.056249839 0.0004531
UK:2-AU:2 -0.007401115 -0.11728511  0.102482875 0.9999999
US:2-AU:2 -0.124798936 -0.23615795 -0.013439927 0.0160203
AU:3-AU:2 -0.117061084 -0.24491470  0.010792533 0.1010228
UK:3-AU:2  0.039234125 -0.08861949  0.167087742 0.9879992
US:3-AU:2 -0.158650640 -0.28315380 -0.034147483 0.0030911
US:2-UK:2 -0.117397821 -0.22728181 -0.007513830 0.0267150
AU:3-UK:2 -0.109659969 -0.23623094  0.016910999 0.1464187
UK:3-UK:2  0.046635240 -0.07993573  0.173206208 0.9626101
US:3-UK:2 -0.151249525 -0.27443515 -0.028063898 0.0051941
AU:3-US:2  0.007737852 -0.12011576  0.135591469 0.9999999
UK:3-US:2  0.164033061  0.03617944  0.291886678 0.0027992
US:3-US:2 -0.033851704 -0.15835486  0.090651454 0.9946340
UK:3-AU:3  0.156295209  0.01384429  0.298746133 0.0202249
US:3-AU:3 -0.041589556 -0.18104118  0.097862065 0.9900401
US:3-UK:3 -0.197884765 -0.33733639 -0.058433144 0.0005514
```

Appendix B

Exploiting homophily for better item recommendations

B.1 CF parameter selection

Selection of optimal parameters for the collaborative filtering, SOC and SCF algorithms was needed. We performed the parameter selection for each dataset separately.

CF based algorithms

We first optimized the two parameters for the UU and II algorithms – the number of neighbors and the minimum overlap. The optimization was performed by brute force exploration of the parameter space and measuring the F1 performance on the validation set. The number of items withheld per weak user for validation was $n_{val}^{Y!360} = 2$ and $n_{val}^{Flickr} = 10$. The best performing algorithms in terms of F1 metric were UU(70,1) and II(100,2) for Yahoo!360 and UU(70,1), II(100,3) for Flickr.

SOC algorithm

For the social popularity algorithm, SOC, we need to optimize two different parameters: the neighborhood size, k and choose the weighting method, $t =$'constant' or 'distance'. Table B.1 shows the effect of parameter variations on F1 performance. Type 'distance' clearly outperforms 'const'. Based on the results on validation set we selected SOC(200,'distance') for Yahoo!360 and SOC(60,'distance') for Flickr.

Yahoo!360				Flickr			
'distance'		'const'		'distance'		'const'	
n	F1	n	F1	n	F1	n	F1
200	0.00764	110	0.00717	60	0.00279	40	0.00248
300	0.00753	200	0.00707	70	0.00279	50	0.00235
110	0.00748	100	0.00707	40	0.00271	70	0.00226
100	0.00744	80	0.00707	50	0.00271	60	0.00226
90	0.00741	90	0.00705	90	0.00262	80	0.00222
80	0.00737	300	0.00703	100	0.00253	110	0.00217
400	0.00734	70	0.00694	80	0.00248	200	0.00217
70	0.00721	60	0.00691	110	0.00235	90	0.00217
500	0.00721	50	0.00676	200	0.00217	100	0.00213
60	0.00700	400	0.00671	300	0.00208	300	0.00177
50	0.00685	40	0.00658	500	0.00195	400	0.00169
40	0.00660	500	0.00658	400	0.00195	500	0.00169

Table B.1: F1 metric results of the SOC algorithm on the validation dataset as a function of number of neighbors n and weighting type t.

Yahoo!360				Flickr			
α_{II}	F1	α_{UU}	F1	α_{II}	F1	α_{UU}	F1
0.05	0.00893	0.3	0.00864	0.2	0.00324	0.01	0.00421
0.1	0.00891	0.1	0.00864	0.05	0.00324	0.1	0.00412
0.01	0.00891	0.05	0.00861	0.1	0.00324	0.05	0.00412
0.3	0.00886	0.2	0.00861	0.01	0.00319	0.3	0.00412
0.2	0.00882	0.4	0.00859	0.3	0.00315	0.4	0.00408
0.4	0.00880	0.01	0.00859	0.4	0.00302	0.2	0.00399
0.5	0.00866	0.5	0.00857	0.5	0.00297	0.6	0.00395
0.6	0.00834	0.6	0.00848	0.6	0.00266	0.5	0.00386
0.7	0.00801	0.7	0.00832	0.7	0.00262	0.7	0.00368
0.8	0.00767	0.8	0.00805	0.8	0.00217	0.8	0.00359
0.9	0.00703	0.9	0.00789	0.9	0.00204	0.9	0.00324
0.95	0.00658	0.95	0.00776	0.99	0.00204	0.95	0.00297
0.99	0.00626	0.99	0.00766	0.95	0.00195	0.99	0.00284

Table B.2: SCF performance comparison for weighted sum voting ('wsum') and different values of α on Yahoo!360 and Flickr validation datasets.

SCF algorithm

We used the optimum collaborative filtering and social networking parameters as described above. We combined each collaborative filtering algorithm with the social algorithm using weighted sum, 'wsum', combination rule with different values of α. This parameter determines the relative weights that the CF component and social component of the algorithm get. We selected the value for each dataset for which F1 performance was best. Performance for various values of α is shown in Table B.2. We selected $\alpha_{UU} = 0.3$ and $\alpha_{II} = 0.05$ for Yahoo!360 and $\alpha_{UU} = 0.01$, $\alpha_{II} = 0.2$ for Flickr.

B.2 Random recommender and dataset size

Let I be the number of all items in dataset, N the number of recommended items (size of the top-N list), and w the number of withheld items. Then we have the following probabilities for a random item i:

$$p(i \text{ is relevant}) = \frac{w}{I} \tag{B.1}$$

$$p(i \in \text{top-N}) = \frac{N}{I} \tag{B.2}$$

For a random recommender the precision equals the probability that a recommended item is relevant and recall equals the probability that a (relevant) item is recommended:

Precision and recall are then given as

$$prec = p(i \text{ is relevant}) = \frac{w}{I} \tag{B.3}$$

$$recall = p(i \in \text{top-N}) = \frac{N}{I} \tag{B.4}$$

Substituting precision and recall into the F1 formula we get:

$$F1 = \frac{2 \cdot prec \cdot recall}{prec + recall} \tag{B.5}$$

$$F1 = \frac{2Nw}{I(N + w)} \tag{B.6}$$

In Equation B.6 we see that absolute value of the F1 metric for random recommender decreases with increasing number of items (I). This effect can be partially compensated by increasing the number of withheld items w and

number of recommended items N. However, the number of withheld items is limited by number of items in each user's purchase history.

In our experiments we use $N = 100$ and $w = 10$. If we compare the parameters of our experiments on the Flickr dataset ($N = 100$, $w = 10$, $I = 200K$) to those of Deshpande [59] performed on MovieLens dataset (N=10, w=1, and I=1682), we would expect the F1 values of a random recommender to be approximately twenty times lower on the larger Flickr dataset than on MovieLens.

These effects of dataset size mean that results on different datasets are not directly comparable. However, the metrics based on partial lists of relevant items are still valuable for comparison of relative performance of individual algorithms.

Index

Bibliography

[1] Jester, online joke recommender system and dataset, http://www.ieor.berkeley.edu/ goldberg/jester-data/, 2001.

[2] Movielens dataset, http://www.grouplens.org/data/, as of 2003.

[3] H Abdi. *Bonferroni and Sidak corrections for multiple comparisons*. Sage, Thousand Oaks, CA, 2007.

[4] Accenture. Leadership in customer service: New expectations, new experiences, the government executive series. Technical report, 2005.

[5] Lada A. Adamic, Orkut Buyukkokten, and Eytan Adar. A social network caught in the web. *First Monday*, 8(6), 2003.

[6] Lada A. Adamic and Bernardo A. Huberman. The web's hidden order. *Communications of the ACM*, 44(9):55–60, 2001.

[7] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. pages 487–499, 1994.

[8] R Albert, H Jeong, and AL Barabasi. Internet: Diameter of the world-wide web. *Nature*, page 130, 1999.

[9] Reid Andersen and Kevin J. Lang. Communities from seed sets. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 223–232, New York, NY, USA, 2006. ACM Press.

[10] Arxiv e-print archive, http://arxiv.org/, as of 2003.

[11] Peter Bailey, Nick Craswell, and David Hawking. Dark matter on the web. page 2. In Poster Proceedings, 9th World-Wide Web Conference, 2000.

[12] Albert-Laszlo Barabasi. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003.

[13] J. A. Barnes. Class and Committees in a Norwegian Island Parish. *Human Relations*, 7(1):39–58, 1954.

[14] Susan Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 2006.

[15] V. Batagelj and A. Mrvar. *Pajek Analysis and Visualization of Large Networks*. Springer, 2003.

[16] Michael Batty. Citation geography: It's about location. *The Scientist*, 17(16):10, 2003.

[17] Michael Batty. The geography of scientific citation. *Environment and Planning A*, 35:761–770, 2003.

[18] Krishna Bharat, Bay-Wei Chang, Monika Rauch Henzinger, and Matthias Ruhl. Who links to whom: Mining linkage between web sites. In *ICDM*, pages 51–58, 2001.

[19] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 11–18, 2004.

[20] I. Bhattacharya and L. Getoor. A Latent dirichlet model for unsupervised entity resolution. *SIAM International Conference on Data Mining*, pages 47–58, 2006.

[21] K.D. Bollacker, S. Lawrence, and C.L. Giles. A system for automatic personalized tracking of scientific literature on the Web. *Proceedings of the fourth ACM conference on Digital libraries*, pages 105–113, 1999.

[22] Kurt D. Bollacker, Steve Lawrence, and C. Lee Giles. Citeseer: an autonous web agent for automatic retrieval and identification of interesting publications. pages 116–123, 1998.

[23] Philip Bonhard, Clare Harries, John McCarthy, and M. Angela Sasse. Accounting for taste: using profile similarity to improve recommender systems. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1057–1066, New York, NY, USA, 2006. ACM Press.

[24] Philip Bonhard and MA Sasse. I thought it was terrible and everyone else loved it – a new perspective for effective recommender systems.

People and Computers XIX - Proceedings of HCI 2005, pages 251–266, 2005.

[25] Craig Boutilier, Richard S. Zelmel, and Benjamin Marlin. Active collaborative filtering. In *Nineteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 98–106, 2002.

[26] Ulrik Brandes and Thomas (Eds.) Erlebach. *Network Analysis*, volume 3418.

[27] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[28] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[29] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 309–320, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

[30] Lukas Brozovsky and Vaclav Petricek. Recommender system for online dating service. In *Proceedings of Znalosti 2007 Conference*, Ostrava, 2007. VSB.

[31] Mauro Brunato and Roberto Battiti. A location-dependent recommender system for the web. In *Proceedings of the MobEA Workshop*, page 5, 2002.

[32] R. Burke. Knowledge-based recommender systems. In *In A. Kent (ed.), Encyclopedia of Library and Information Systems*, page 23, New York, 2000. Marcel Dekker.

[33] R. Burke, M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin and. Integrating knowledge-based and collaborative-filtering recommender systems. In *Proceedings of the Workshop on AI and Electronic Commerce. AAAI 99*, page 4, Orlando, Florida, 1999.

[34] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[35] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.

[36] Tsallis C and de Albuquerque MP. Are citations of scientific papers a case of nonextensivity? *EUROPEAN PHYSICAL JOURNAL B*, 13:777–780, 2000.

[37] Alexandre Caldas. On the web structure and digital knowledge bases. pages 1–18, 2004.

[38] J. Canny. Collaborative filtering with privacy. In *In IEEE Symposium on Security and Privacy*, pages 45–47, May 2002.

[39] John Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, New York, NY, USA, 2002.

[40] Peter J. Carrington, John Scott, and Stanley Wasserman (eds.). *Models and Methods in Social Network Analysis*. Cambridge University Press, New York, 2005.

[41] Demchak C.C., Friis C., and La Porte T.M. Webbing governance: National differences in constructing the public face. *in G.D.Garson (ed.) Handbook of Public Information Systems*, 2000.

[42] S. Chakrabarti, M. van den Berg, and B. Dom. Focused Crawling: A New Approach for Topic-Specific Resource Discovery. *WWW Conference*, pages 545–562, 1999.

[43] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 65–74, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[44] Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the Web's link structure. *Computer*, 32(8):60–67, 1999.

[45] Mao Chen and Jaswinder Pal Singh. Computing and using reputations for internet ratings. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 154–162, New York, NY, USA, 2001. ACM Press.

[46] Heyning Adrian Cheng. Knowledgescapes: A probabilistic model for mining tacit knowledge for information retrieval. Technical report, 2001.

[47] J. Cho, H. Garcia-Molina, and L. Page. Efficient Crawling Through URL Ordering. *WWW7 / Computer Networks*, 30(1-7):161–172, 1998.

[48] N. Christianini and J. Shawe-Taylor. *An introduction to support vector machines: and other kernel-based learning methods*. CAMBRIDGE UNIV PR, 2000.

[49] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, , and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, page 8, August 1999.

[50] Compuscience database, http://www.zblmath.fiz-karlsruhe.de/COMP/quick.htm, as of 2003.

[51] The Thomson Corporation. How do we identify highly cited researchers?, as of 2003.

[52] CoRR, http://xxx.lanl.gov/archive/cs/, as of 2003.

[53] D. Cosley, S.K. Lam, I. Albert, J. Konstan, and J. Riedl. Is seeing believing? how recommender systems influence users' opinions. pages 585–592. In Proceedings of CHI 2003 Conference on Human Factors in Computing Systems, 2003.

[54] Dan Cosley, Steve Lawrence, and David M. Pennock. REFEREE: An open framework for practical testing of recommender systems using researchindex. In *28th International Conference on Very Large Databases, VLDB 2002*, pages 35–46, Hong Kong, August 20–23 2002.

[55] Cs bibtex database, http://liinwww.ira.uka.de/bibliography/, as of 2003.

[56] DBLP, http://dblp.uni-trier.de/, as of 2003.

[57] Dennis DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 249–256, Pittsburgh, Pennsylvania, 2006. ACM Press.

[58] Ayhan Demiriz. Enhancing product recommender systems on sparse binary data. *Data Min. Knowl. Discov.*, 9(2):147–170, 2004.

[59] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[60] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, and Marco Gori. Focused crawling using context graphs. In *26th International Conference on Very Large Databases, VLDB 2000*, pages 527–534, Cairo, Egypt, 10–14 September 2000. VLDB.

[61] Robert P. W. Duin and David M. J. Tax. Experiments with classifier combining rules. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 16–29, London, UK, 2000. Springer-Verlag.

[62] Patrick F. Dunn. *Measurement and Data Analysis for Engineering and Science*. McGraw-Hill, New York, 2005.

[63] W.H. Dutton, C. di Genarro, and Millwood. *A. The internet in Britain: The Oxford internet Survey (OxIS)*. Hargrave, 2005.

[64] N. Elmqvist and P. Tsigas. CiteWiz: A Tool for the Visualization of Scientific Citation Networks. *Department of Computing Science, Chalmers University of Technology, Technical report CS*, pages 1–22, 2004.

[65] P. Erdos and A. Renyi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[66] Tobias Escher, Helen Margetts, Vaclav Petricek, and Ingemar J. Cox. Governing from the centre? comparing the nodality of digital governments. In *the 2006 Annual Meeting of the American Political Science Association*, 2006.

[67] Brian S. Everitt and Torsten Hothorn. *A Handbook of Statistical Analyses Using R*. Chapman and Hall/CRC, Boca Raton, FL, USA, 2006.

[68] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[69] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[70] O. Frank and D. Strauss. Markov Graphs. *Journal of the American Statistical Association*, 81(395):832–842, 1986.

[71] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS – clustering categorical data using summaries. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 73–83, 1999.

[72] E. Garfield. The impact factor. *Current Contents*, 25(20):3–7, 1994.

[73] Cap Gemini. *Online Availability of Public Services: How is Europe Progressing? Web-based survey on electronic public services, Report of the fifth measurement*. (European Commission Directorate General for Information Society and Media), October 2004.

[74] Andreas Geyer-Schulz and Michael Hahsler. Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory. In Brij Masand, Myra Spiliopoulou, Jaideep Srivastava, and Osmar R. Zaiane, editors, *Fourth WebKDD Workshop: Web Mining for Usage Patterns & User Profiles*, pages 100–114, Edmonton, Canada, July 2002.

[75] Andreas Geyer-Schulz, Michael Hahsler, and Maximillian Jahn. Educational and scientific recommender systems: Designing the information channels of the virtual university. *International Journal of Engineering Education*, 17(2):153–163, 2001.

[76] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *UK Conference on Hypertext*, pages 225–234, 1998.

[77] CL Giles, KD Bollacker, and S Lawrence. Citeseer: an automatic citation indexing system. *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998.

[78] Lorraine Mc Ginty and Barry Smyth. Deep dialogue vs casual conversation. In *Proceedings of the Workshop on Personalization in eCommerce*

at the Second International Conference on Adaptive Hypermedia and Web-Based Systems (AH-02), pages 80–89, 2001.

[79] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the. ACM*, 35(12):61–70, 1992.

[80] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[81] N. Good, J.B. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99)*, pages 439–446, 1999.

[82] Good university guide, The Times, 2002.

[83] I. Graafland-Essers and E. Ettedgui. Benchmarking e-government in europe and the u.s. Technical report, 2003.

[84] J. Graupmann, R. Schenkel, and G. Weikum. The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents. *Proceedings of the 31st international conference on Very large data bases*, pages 529–540, 2005.

[85] R. Guha. Open rating systems. Technical report, Stanford University, CA, USA, 2003.

[86] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587. Morgan Kaufmann, 2004.

[87] P Haase, R Siebes, and F van Harmelen. Peer selection in peer-to-peer networks with semantic topologies. *Proceedings of the International Conference on Semantics in a ...*, pages 108–125, 2004.

[88] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on*, pages 296–305, 2004.

[89] H. Han, H. Zha, and C.L. Giles. Name disambiguation in author citations using a K-way spectral clustering method. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 334–343, 2005.

[90] Hui Han, C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. Automatic document metadata extraction using support vector machines. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 37–48, Washington, DC, USA, 2003. IEEE Computer Society.

[91] F Harary. *Graph theory*. Addison-Wesley, Reading, Mass, 1969.

[92] S. Harnad and L. Carr. Integrating, navigating, and analyzing open eprint archives through open citation linking (the OpCit project). *Current Science*, 79(5):629–638, 2000.

[93] J. Hassell, B. Aleman-Meza, and I.B. Arpinar. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text.

[94] Conor Hayes, Paolo Massa, Paolo Avesani, and Padraig Cunningham. An on-line evaluation framework for recommender systems. Technical report, 2002.

[95] X. He, H. Zha, C. Ding, and H. Simon. Web document clustering using hyperlink structures. Technical Report CSE-01-006, Department of Computer Science and Engineering, Pennsylvania State University., 2001.

[96] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM Press.

[97] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[98] UK Research Assessment Exercise. *Higher Education and Research Opportunities*, 2002.

[99] Matthew Hindman, Kostas Tsioutsiouliklis, and Judy A. Johnson. Googlearchy: How a few heavily-linked sites dominate politics on the

web. In *Annual Meeting of the Midwest Political Science Association*, June 03 2003.

[100] T. Hofmann. Latent semantic models for collaborative filtering. *ACM TOIS*, 22(1):89–115, Jan 2004.

[101] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *IJCAI*, pages 688–693, 1999.

[102] PW Holland and S. Leinhardt. Exponential Family of Probability Distributions for Directed Graphs. *J. AM. STAT. ASSN.*, 76(373):33–50, 1981.

[103] Y. Hong, B.W. On, and D. Lee. System Support for Name Authority Control Problem in Digital Libraries: OpenDBLP Approach. *European Conf. on Digital Libraries (ECDL), Bath, UK, Sep*, 2004.

[104] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 541–546, 2003.

[105] BA Huberman and LA Adamic. Growth dynamics of the world-wide web. *J. Reprod. Fertil*, page 131, 1993.

[106] Bernardo A. Huberman, Peter L. T. Pirolli, James E. Pitkow, and Rajan M. Lukose. Strong regularities in World Wide Web surfing. *Science*, 280(5360):95–97, 1998.

[107] L. J. Hubert and J. Schultz. Quadratic assignment as a general data analysis strategy. *British Journal of Mathematical and Statistical Psychology*, (29):190–241, 1976.

[108] P. Ingversen. The calculation of web impact factors. *Journal of Documentation, 4(2)*, pages 236–243, 1998.

[109] Melody Y. Ivory, Rashmi R. Sinha, and Marti A. Hearst. Empirically validated web page design metrics. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 53–60, New York, NY, USA, 2001. ACM Press.

[110] H. Jiang, W. Lou, and W. Wang. Three-tier Clustering: an Online Citation Clustering System. *Proceedings of the Second international Conference on Web-Age Information Management (WAIM2001)*, pages 237–248, 2001.

[111] Jussi Karlgren. Newsgroup clustering based on user behavior – a recommendation algebra. Technical Report T94:04, Stockholm, Sweden, 1994.

[112] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254, New York, NY, USA, 2001. ACM Press.

[113] Henry Kautz, Bart Selman, and Mehul Shah. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[114] M.M. Kessler. *Bibliographic Coupling Between Scientific Papers*. Massachusetts Institute of Technology, 1962.

[115] J. Kim and G. Fox. Scalable Hybrid Search on Distributed Databases. *Proceedings of International Workshop of Autonomic Distributed Data and Storage Systems Management (To appear)*.

[116] J. Kim and G. Fox. A Hybrid Keyword Search across Peer-to-Peer Federated Databases. *Proceedings of East-European Conference on Advances in Databases and Information Systems (ADBIS), September*, 2004.

[117] Mee-Jean Kim. Comparative study of citations from papers by korean scientists and their journal attributes. *Journal of Information Science*, 24:113–121, 1998.

[118] M Kinateder and K Rothermel. Architecture and algorithms for a distributed reputation system. *Proceedings of the First International Conference on Trust Management*, pages 1–16, 2003.

[119] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

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

[121] S. Klink, M. Ley, E. Rabbidge, P. Reuther, B. Walter, and A. Weber. Browsing and visualizing digital bibliographic data. *Symposium on Visualization*, pages 19–21, May 2004.

[122] Janne S. Kotiaho. Papers vanish in mis-citation black hole. *Nature*, 398:19, 1999.

[123] Janne S. Kotiaho. Unfamiliar citations breed mistakes. *Nature 400*, 400:307, jul 1999.

[124] H. Krottmaier. Automatic Support in the Review Process. *Proceedings of the Workshops of The First Eurasian Conference on Advances in Information and Communication Technology (EurAsia-ICT 2002)*, pages 467–471, 2002.

[125] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, 1999.

[126] J. Laherrre and D. Sornette. Stretched exponential distributions in nature and economy: 'fat tails' with characteristic scales. *The European Physical Journal B - Condensed Matter*, 2(4):525–539, 1998.

[127] Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, pages 393–402, New York, NY, USA, 2004. ACM Press.

[128] Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 393–402, New York, NY, USA, 2004. ACM Press.

[129] S. Lawrence, K. Bollacker, and C.L. Giles. Autonomous citation matching. *Proceedings of the Third International Conference on Autonomous Agents*, pages 392–393, 1999.

[130] S. Lawrence, K. Bollacker, and C.L. Giles. Indexing and retrieval of scientific literature. *Proceedings of the eighth international conference on Information and knowledge management*, pages 139–146, 1999.

[131] Steve Lawrence. Online or invisible? *Nature*, 411(6837):521, 2001.

[132] Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.

[133] M.L. Lee, T.W. Ling, and W.L. Low. Designing functional dependencies for XML. *Proceedings of the 8th International Conference on Extending Database Technology*, pages 124–141, 2002.

[134] S. Lehmann, B. Lautrup, and A. D. Jackson. Citation networks in high energy physics. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 68(2):026113, 2003.

[135] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining (SDM'05)*, 2005.

[136] M. Ley and P. Reuther. Maintaining an online bibliographical database: The problem of data quality. In *In proceedings of the EGCS 2006*, 2006.

[137] Michael Ley. Dblp: A www bibliography on databases and logic programming, 1997.

[138] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing*, 7(1):76–80, 2003.

[139] W. Lu, J. Janssen, E. Milios, and N. Japkowicz. Node similarity in networked information spaces. *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, pages 1–23, 2001.

[140] LEY M. The dblp computer science bibliography: Evolution, research issues, perspectives. *In SPIRE*, pages 1–10, 2002.

[141] B. Marlin. Collaborative filtering: A machine learning perspective. Master's thesis, 2004.

[142] C W Matthiessen and AW Schwarz. Scientific centres in europe. *Urban Studies 36*, pages 453–477, 1999.

[143] R. M. May. The scientific wealth of nations. *Science 275*, pages 793–795, 1997.

[144] David W. McDonald. Recommending collaboration with social networks: a comparative evaluation. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 593–600, New York, NY, USA, 2003. ACM Press.

[145] Amy McGovern, Lisa Friedland, Michael Hay, Brian Gallagher, Andrew Fast, Jennifer Neville, and David Jensen. Exploiting relational structure to understand publication patterns in high-energy physics. *SIGKDD Explor. Newsl.*, 5(2):165–172, 2003.

[146] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[147] Stuart E. Middleton. Exploiting synergy between ontologies and recommender systems. In *Semantic web conference, WWW2002*, page 10, 2002.

[148] MO'Connor and J Herlocker. Clustering items for collaborative filtering. *the Proceedings of SIGIR-2001 Workshop on Recommender Systems*, page 4, 2001.

[149] M. Montaner, B. Lopez, and J. Rosa. Opinionbased filtering through trust. In *Proceedings of the 6th International Workshop on Cooperative Information Agents VI*, 2002.

[150] Miquel Montaner, Beatriz Lopez, and Josep Lluis De La Rosa. A taxonomy of recommender agents on the internet. *Artif. Intell. Rev.*, 19(4):285–330, 2003.

[151] Microsoft anonymous web data, http://kdd.ics.uci.edu/databases/msweb/msweb.html, 1998.

[152] D. Neves and F. Adrian. Stepping Stones and Pathways: Improving Retrieval by Chains of Relationships between Documents.

[153] M. E. J. Newman. The structure of scientific collaboration networks. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 98, pages 404–409, 2001.

[154] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[155] A.Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14(2):849–856, 2001.

[156] J. Nielsen. *Usability Engineering.* Academic Press Inc., US, 1994.

[157] W. Nooy, A. Mrvar, and Batagelj V. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, 2005.

[158] Networked computer science technical reference library, http://www.ncstrl.org/, as of 2003.

[159] University of Illinois Library. Graduate and research program rankings, 2002.

[160] Michael O'Mahony, Neil Hurley, Nicholas Kushmerick, and Guenole Silvestre. Collaborative recommendation: A robustness analysis. *ACM Trans. Inter. Tech.*, 4(4):344–377, 2004.

[161] Michael P. O'Mahony, Neil Hurley, and Guenole C. M. Silvestre. Promoting recommendations: An attack on collaborative filtering. In *DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications*, pages 494–503, London, UK, 2002. Springer-Verlag.

[162] B.W. On, D. Lee, J. Kang, and P. Mitra. Comparative study of name disambiguation problem using a scalable blocking-based framework. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 344–353, 2005.

[163] Andy Oram, editor. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly, 2001.

[164] A. O. Oswald. A crisis of quality. *Education Guardian 15 November*, 2002.

[165] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[166] M Papagelis, D Plexousakis, and T Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. *Proceedings of the 3rd International Conference on Trust Management*, pages 224–239, 2005.

[167] Han Woo Park. Hyperlink network analysis: A new method for the study of social structure on the web. In *SUNBELT*, pages 49–61, 2003.

[168] Seung-Taek Park, Alexy Khrabrov, David M. Pennock, Steve Lawrence, C. Lee Giles, and Lyle H. Ungar. Static and dynamic analysis of the internet's susceptibility to faults and attacks. In *INFOCOM*, 2003.

[169] Seung-Taek Park, David Pennock, Omid Madani, Nathan Good, and Dennis DeCoste. Naive filterbots for robust cold-start recommendations. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 699–705, New York, NY, USA, 2006. ACM Press.

[170] Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5 - 6):393–408, December 1999.

[171] F. Peng and S.S. Chawathe. XSQ: A streaming XPath engine. *ACM Transactions on Database Systems (TODS)*, 30(2):577–623, 2005.

[172] David Pennock, Eric Horvitz, Steve Lawrence, and C. Lee Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI 2000*, pages 473–480, Stanford, CA, 2000.

[173] Y. Petinot, C.L. Giles, V. Bhatnagar, P.B. Teregowda, H. Han, and I. Councill. A service-oriented architecture for digital libraries. *Proceedings of the 2nd international conference on Service oriented computing*, pages 263–268, 2004.

[174] Y. Petinot, C.L. Giles, V. Bhatnagar, P.B. Teregowda, H. Han, and I. Councill. CiteSeer-API: towards seamless resource location and interlinking for digital libraries. *Proceedings of the Thirteenth ACM conference on Information and knowledge management*, pages 553–561, 2004.

[175] Vaclav Petricek, Ingemar J. Cox, Hui Han, Isaac Councill, and C. Lee Giles. A comparison of on-line computer science citation databases. In *9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005)*, pages 438–449. Springer, 2005.

[176] Vaclav Petricek, Ingemar J. Cox, Hui Han, Isaac G. Councill, and C. Lee Giles. Modeling the author bias between two on-line computer science citation databases. In *WWW '05: Special interest tracks and*

posters of the 14th international conference on World Wide Web, pages 1062–1063, New York, NY, USA, 2005. ACM Press.

[177] Vaclav Petricek, Tobias Escher, Ingemar J. Cox, and Helen Margetts. The web structure of e-government - developing a methodology for quantitative evaluation. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 669–678, New York, NY, 2006. ACM Press.

[178] Vaclav Petricek, Seoung-Taek Park, and Ingemar J. Cox. Socially informed collaborative filtering. In *in review*, 2007.

[179] Alexandrin Popescul, Lyle Ungar, David Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *17th Conference on Uncertainty in Artificial Intelligence*, pages 437–444, Seattle, Washington, August 2–5 2001.

[180] Todd M. La Porte, Chris C. Demchak, and Christian Friis. Webbing governance: global trends across national-level public agencies. *Communications of the ACM*, 44(1):63–67, January 2001.

[181] D. De Solla Price. *Little Science, Big Science*. Columbia University Press, New York, NY, 1963.

[182] Al Mamunur Rashid, George Karypis, and John Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. *Proceedings of SIAM International Conference on Data Mining*, 2005.

[183] Kirsten Swearingen Rashmi. Interaction design for recommender systems. In *Proceedings of DIS2002*, page 10, London, UK, 2002. ACM Press.

[184] S. Redner. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B*, 4:131–134, 1998.

[185] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. pages 713–719, 2005.

[186] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the. ACM*, 40(3):56–58, 1997.

[187] FJ Richards. A flexible growth function for empirical use. *J. Exp. Bot*, 10(29):290–300, 1959.

[188] E. Rogers. *Diffusion of innovations (4th ed.)*. Free Press, 1995.

[189] G Salton and J Michael McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[190] B Sarwar, G Karypis, J Konstan, and J Reidl. Item-based collaborative filtering recommendation algorithms. *Proceedings of the tenth international conference on World Wide Web*, pages 285–295, 2001.

[191] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems–a case study. page 12, 2000.

[192] B Sarwar, G Karypis, J Konstan, and J Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. *Fifth International Conference on Computer and Information Science*, 2002.

[193] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–167, 2000.

[194] J. Ben Schafer, Joseph A. Konstan, and John Riedi. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999.

[195] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, 2001.

[196] Scientific citation index, http://www.isinet.com/products/citation/sci/, as of 2003.

[197] Sciencedirect digital library, http://www.sciencedirect.com, as of 2003.

[198] P.O. Seglen. Why the impact factor of journals should not be used for evaluating research, 1997.

[199] G. Shani, D. Heckerman, and R.I. Brafman. An MDP-Based Recommender System. *The Journal of Machine Learning Research*, 6:1265–1295, 2005.

[200] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating word of mouth. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[201] A. Sidiropoulos and Y. Manolopoulos. A new perspective to automatically rank scientific conferences using digital libraries. *Information Processing and Management*, 41(2):289–312, 2005.

[202] M. V. Simkin and V. P. Roychowdhury. Read before you cite! *COMPLEX SYST.*, 14:269, 2003.

[203] Rashmi R. Sinha and Kirsten Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, page 6, 2001.

[204] T.A.B. Snijders, P. Pattison, G.L. Robins, and M. Handcock. New specifications for exponential random graph models. *Sociological Methodology*, 2006.

[205] Taylor Nelson Sofres. Government online: An international perspective. Technical report, 2003.

[206] Spires high energy physics literature database, http://www.slac.stanford.edu/spires/hep/.

[207] NEC Steve Lawrence and NEC Kurt Bollacker. Digital Libraries and Autonomous Citation Indexing. *Contact*, 32:67–71, 1999.

[208] Martin Svensson, Jarmo Laaksolahti, Kristina Hk, and Annika Waern. A recipe based on-line food store. In *Intelligent User Interfaces*, pages 260–263, 2000.

[209] Y.F. Tan, M.Y. Kan, and D. Lee. Search engine driven author disambiguation. *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 314–315, 2006.

[210] Mike Thelwall. Conceptualizing documentation on the web: an evaluation of different heuristic-based models for counting links between university web sites. *Journal of the American Society for Information Science and Technology*, 53(12):995–1005, 2003.

[211] Roberto Torres, Sean M. McNee, Mara Abel, Joseph A. Konstan, and John Riedl. Enhancing digital libraries with techlens. pages 228–236. JCDL, 2004.

[212] Hiroyuki Uchiyama, Makoto Onizuka, and Takashi Honishi. Distributed xml stream filtering system with high scalability. *icde*, 0:968–977, 2005.

[213] L. Ungar and D. Foster. Clustering methods for collaborative filtering. 1998.

[214] UNPAN. United nations world public sector report 2003: E-government at the crossroads department of economics and social affairs. Technical report, 2003.

[215] Shardanand Upendra. Social information filtering for music recommendation, s.m. thesis, program in media arts and sciences, massachusetts institute of technology. Master's thesis, 1994.

[216] Tom Valente. Network models and methods for studying the diffusion of innovations. pages 98–116, 2005.

[217] T.W. Valente, J.B. Unger, and C.A. Johnson. Do popular students smoke? The association between popularity and smoking among middle school students. *Journal of Adolescent Health*, 37(4):323–329, 2005.

[218] Marijtje A. J. van Duijn, Tom A. B. Snijders, and Bonne J. H. Zijlstra. p2: a random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58(2):234–254, 2004.

[219] Vladimir Vapnik and S.Kotz. *Estimation of Dependences Based on Empirical Data*. Springer, 2006.

[220] Alexei Vazquez. Statistics of citation networks. pages 1–12, 2001.

[221] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM Press.

[222] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[223] Duncan J. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, February 2004.

[224] Jianshu Weng, Chunyan Miao, and Angela Goh. Improving collaborative filtering with trust-based metrics. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1860–1864, New York, NY, USA, 2006. ACM Press.

[225] Jianshu Weng, Chunyan Miao, Angela Goh, and Dongtao Li. Trust-based collaborative filtering. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 299–300, New York, NY, USA, 2005. ACM Press.

[226] D. West. *Digital Government: Technology and Public Sector Performance*. Princeton University Press, 2005.

[227] H. D. White and K. W. McCain. *Bibliometrics*, volume 24. Elsevier, Amsterdam, 1989.

[228] B.B. Yao, M.T. Ozsu, and J. Keenleyside. XBench-A Family of Benchmarks for XML DBMSs. *Proceedings of the VLDB 2002 Workshop EEXTT and CAiSE 2002 Workshop DTWeb on Efficiency and Effectiveness of XML Tools and Techniques and Data Integration over the Web-Revised Papers*, pages 162–164, 2003.

[229] K. Yu, X. Xu, J. Tao, M. Ester, and H. Kriegel. Instance selection techniques for memory-based collaborative filtering. In *SDM '02*, page 16, 2002.

[230] S Zhou and RJ Mondragon. Accurately modeling the internet topology. *Physical Review E*, (066108):1–8, 2004.

[231] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM Press.