Information Retrieval Models

by

MOHAMMAD ALI NOROZI

THESIS
for the degree of

MASTER OF SCIENCE

(Master i Anvendt matematikk og mekanikk)

Faculty of Mathematics and Natural Sciences
University of Oslo

May 2008

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo
To the memory of my parents, my family and Fast Search & Transfer
Acknowledgements

Hard work is thought to be difficult but once you get used to work hard you enjoy — the hard work. Because its first rewarding than challenging.

To keep a young and energetic soul alive and young is not an individual task. It is the whole that makes it possible to see the end. The road has been rather long — not to mention somewhat winding.

To dive into the sea of concepts and ideas and to come out and bring about the precious treasures of new and peculiar ideas, that make a difference in the area, is a really challenging task — but if done with interest and enthusiasm.

At times you feel that the work that you are doing have no value, but there are few times that you feel your work is going to be worth. Overall its a balance between what you do and what you think is worth, and the overall work in the thesis is to find that balance where you feel contended with work. I think I have found my satisfaction to a certain level before I hand in the thesis.

The one year long thesis has been a motivating and has been my good fortune to encounter many people who have given me more of their time, companionship, professional and personal help, and above all: patience than was perhaps warranted by my seeming determination to indefinitely position the deadline for finishing this thesis at “next year”.

I would first of all like to thank my supervisor at FAST, Torbjørn Helvik. He not only gave me the scientific support and supervision that a graduate student can expect from his supervisor, but he also allowed and encouraged me to remain up to date and energetic. Thanks to him, I have never been without a desk, a computer, a friendly ear, and the occasional job at the Fast Search & Transfer. Without those I would never have made it this far.

Geir Dahl, of University of Oslo, is my primary supervisor at University. His ideas, his research, and especially his unique brand of enthusiasm form the bedrock on which much of this thesis was built.

Torgeir Hovden, of Fast Search & Transfer, is my second supervisor at Fast Search & Transfer. I could not have wished for a more thorough discussion partner and sounding board (on any subject under the sun), while his excellent long term views (particularly the managerial perspective and the broader and prospective views) has been instrumental in helping to be within the boundaries of thesis and remain dynamic.

A list that, alas, has far too many names on it to mention separately is that of all the co-workers, group members, and roommates that I have worked, talked, and lunched with over the year. My gratitude goes out to all these colleagues and fellows at the Fast Search & Transfer; the Computational Scicence (CMA) group at
University of Oslo.

Moving towards more personal acknowledgements, I would like to execute a big blocksend() of aggregated thanks towards all my family and friends — with a special shout-out to my dearest friends Sultana Ali and Sani e Zahra — for their help, friendship and patience, and for the fact that they never gave in to the temptation to make fun of my thesis.

I am, of course, particularly indebted to my parents and family for their monumental, unwavering support and encouragement on all fronts. They have truly always been there for me, and without them none of this would have been even remotely possible.

Oslo,                                          Mohammad Ali Norozi
May 2008
Abstract

In “Information Retrieval”, relevance is a numerical score assigned to a search result, representing how well the results meet the information need of the user that issued the search query. In many cases, a result’s relevance determines the order in which it is presented to the user. In this thesis we have explored the information retrieval models in general and relevancy ranking within information retrieval in particular.

Several mathematical tools have been used in research for improving the relevancy ranking models. A simple yet useful type of relevancy models are based on viewing each document and each query as elements in a high dimensional vector space, and using the angle between the document and the query as a measure of similarity. More advanced concepts in linear algebra, such as the Singular Value Decomposition, and theory of Markov chains have also been employed for innovating relevancy ranking. Some of researches have also suggested and which is also true to certain extent that probability theoretic based models, such as inference and neural networks are the best theoretical foundation for relevancy ranking models.

A particularly important question is how to assess the “goodness” of a relevancy model. There is also a greater need to focus on effective and optimized implementations, such as query latency times should be in the sub-second domain. Theoretically “recall” and “precision” are used as measures for analyzing the effectiveness of a relevancy ranking models. But with the advent of new and sophisticated models there is a need to have a better framework for evaluation.

In collaboration with Fast Search and Transfer ASA (FAST), I have conducted a study in the area of information retrieval and relevancy ranking models. After an initial literature study, I have partially looked into FAST’s query evaluation framework to perform experiments and investigations. But due to unavailability of the required structures that I needed for the study, I had to perform an independent and standalone evaluations.

I have looked into different relevancy models applied in numerous contexts and based on them formulated the focus area, i.e., the Link Analysis Ranking (LAR). The initial investigations therefore made it possible to suggest a case which could be targeted during the study. I have analyzed in light of the theory different concepts and models.

A related area which also has been considered is the automated optimization of the relevancy models. Obtaining good query relevancy is also related to other parts of the system, such as document pre-processing, indexing approaches, the availability of statistics about term distribution, and various query optimization techniques. Thus, a study of such aspects of the system may also be required, in this study we have discussed them, but they are not the focus of this thesis. Instead I looked purely into relevancy models and discussed other important elements only tentatively.
I have found a novel improvement in algorithms like HITS, SALSA and their descendants (e.g., Exponentiated and Randomized HITS) using the *Extrapolation techniques*. Through which I was able to accelerate the algorithms in terms of reducing the number of iterations and therefore uncovered a much faster convergence. In the experiments I even got much better results than theoretically predicted results, a speedup of order $3 - 19$ times better.

The contribution of Extrapolation is unique primarily because I have extensively read through minute details through empirical and theoretical analyses. And it is first time that Extrapolation techniques are used and evaluated in query-dependent algorithms. Previously same kind of techniques has been applied to PageRank, but the improvements were not as promising as in the case of the query-dependent algorithms.

Later, I have theoretically presented numerous personalization models both in query-independent and query-dependent LAR algorithms. And empirically analyzed a smaller subset of the ideas theoretically discussed earlier. A generic framework for personalization has also been discussed which might be considered as an initial step towards personalization.

In the end, extensive experimental evaluations have been performed for the ideas explored earlier in extrapolation and personalization for query-dependent algorithms.
# Contents

Acknowledgements i  
Abstract iii  
List of Figures ix  
List of Tables xi  

1 Introduction  
   1.1 Introduction to Search Engines & Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . . 1  
      1.1.1 A Brief taxonomy of Search . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1  
      1.1.2 Search Engines and its Challenges . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1  
      1.1.3 Elements of search . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4  
      1.1.4 World Wide Web (WWW) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5  
   1.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8  
   1.3 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8  

I Theoretical Background 11  

2 Information Retrieval Models and trends 13  
   2.1 Information Retrieval (IR) Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13  
   2.2 The IR Process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14  
      2.2.1 Document Corpus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14  
      2.2.2 Document Manager . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14  
      2.2.3 Indexer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16  
      2.2.4 Query processing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16  
      2.2.5 Matcher . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17  
      2.2.6 Ranking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17  
      2.2.7 User Feedback . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17  
   2.3 An Overview of the Conventional IR Systems . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
## CONTENTS

2.3.1 Boolean model ................................................................. 18  
2.3.2 Vector model ................................................................. 19  
2.3.3 Probabilistic model ............................................................ 22  
2.4 Future prospects of the classical models .................................. 23  
2.5 Boolean Based Models .......................................................... 24  
2.5.1 Fuzzy set Model .............................................................. 24  
2.5.2 Extended Boolean IR Model ............................................... 24  
2.6 Vector Based Models ........................................................... 25  
2.6.1 Generalized Vector Space model ........................................ 25  
2.6.2 Latent Semantic Indexing (LSI) model ................................. 26  
2.6.3 Neural Network Model ..................................................... 28  
2.7 Probabilistic or network based models ..................................... 29  
2.7.1 Basic concepts ............................................................... 29  
2.7.2 Inference Network Model ................................................ 31  
2.8 Structured and Un-structured document Retrieval Models ................ 36  
2.8.1 Background ................................................................. 36  
2.8.2 Brief Taxonomy ............................................................. 36  
2.8.3 BM25 ................................................................. 37  
2.8.4 Anchor Text ............................................................... 37  
2.8.5 Only Structured Documents ............................................ 37  
2.9 Summary and reflections ..................................................... 38  

### 3 Link Analysis Ranking

3.1 Link Analysis Ranking (LAR) .................................................. 39  
3.2 InDegree Algorithm ............................................................ 40  
3.2.1 Relevance to other Models ............................................. 41  
3.3 PageRank ................................................................. 41  
3.3.1 Web Graph ............................................................... 41  
3.3.2 Google’s PageRank ...................................................... 43  
3.3.3 Link Matrix ............................................................... 44  
3.3.4 Markov chain and Random Surfer (Walk) ............................. 45  
3.3.5 Limitations and Adjustment ............................................ 46  
3.3.6 Power Method ............................................................ 47  
3.3.7 Effects of Random Jump ................................................ 48  
3.3.8 The Algorithm ............................................................ 48  
3.4 HITS (Hypertext Induced Topic Search) .................................... 49  
3.4.1 Notion of Authority ...................................................... 49  
3.4.2 Authorities and Hubs ................................................... 49  
3.4.3 Focused Subgraph ...................................................... 50  
3.4.4 Hub and Authority scores ............................................. 51  
3.4.5 Principal Eigenvectors ................................................ 52  
3.4.6 Non-Unique authority or hub score and Adjustment ............... 53
3.4.7 Random Walks and HITS .................................................. 54
3.4.8 Singular Value Decomposition ......................................... 54
3.4.9 TKC Effect ................................................................. 56
3.4.10 Algorithm ............................................................... 57

3.5 SALSA (Stochastic Approach for Link-Structure Analysis) .......... 57
  3.5.1 Informative and Non-informative links ............................. 58
  3.5.2 Bipartite Graph (Hubs and Authorities) ............................ 58
  3.5.3 Two Random Walks .................................................... 59
  3.5.4 Stochastic Matrices .................................................... 60
  3.5.5 Algorithm ............................................................... 62

II Evaluations, Analyses and Experiments 63

4 Evaluations and Analyses of LAR Models 65
  4.1 Link Analysis Ranking Process ......................................... 65
  4.2 Implicit Properties of HITS algorithm and Problems ................. 66
    4.2.1 Example .............................................................. 67
    4.2.2 Approaches to address the problems ............................ 68
  4.3 Extrapolation Techniques to accelerate the Convergence ............ 70
    4.3.1 Aitken’s $\Delta^2$ Extrapolation ................................. 71
    4.3.2 Quadratic Extrapolation ........................................... 73
    4.3.3 Power ($A^d$) Extrapolation ...................................... 76
    4.3.4 Insights into Extrapolation ....................................... 78
  4.4 Personalization ........................................................... 79
    4.4.1 Intelligent Surfer ................................................... 81
    4.4.2 General Formulation of Personalization .......................... 82
    4.4.3 Approaches towards Personalization .............................. 83
    4.4.4 Personalized and Stabilized HITS ................................. 85

5 Experimental Evaluations 91
  5.1 Experimental Setup ..................................................... 91
    5.1.1 The Graph and the Dataset ....................................... 91
    5.1.2 The Queries ......................................................... 92
    5.1.3 Query Statistics .................................................... 93
    5.1.4 Measures ............................................................. 94
    5.1.5 Convergence .......................................................... 94
    5.1.6 User Study ............................................................ 95
  5.2 Algorithms and Results of Experiments ................................ 95
    5.2.1 Extrapolation ....................................................... 97
    5.2.2 Personalization ..................................................... 109
III Conclusions, Recommendations & Future Study 121

6 Conclusions 123

6.1 Objectives of the study 124

6.1.1 Document ranking strategies 124

6.1.2 Improvements and Contributions 124

6.1.3 Experiments and Evaluations 125

6.2 Future Work 126

6.3 Problems and impediments 127

6.4 Recommendations and Broader Implications 127

Bibliography 129

A Experiments - Extrapolation 135

B Experiments - Top-15 Results 171
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Information Retrieval Process</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Vector model</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>Neural network</td>
<td>28</td>
</tr>
<tr>
<td>2.4</td>
<td>Bayesian Network</td>
<td>30</td>
</tr>
<tr>
<td>2.5</td>
<td>Inference Network</td>
<td>32</td>
</tr>
<tr>
<td>3.1</td>
<td>The webgraph or the internet map - Image taken from [com]</td>
<td>42</td>
</tr>
<tr>
<td>3.2</td>
<td>Network of five pages referencing each other via hyperlink</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Network of five documents containing “dangling nodes”</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>The Base-set</td>
<td>50</td>
</tr>
<tr>
<td>3.5</td>
<td>Connectivity of Web, image taken from [Bro00]</td>
<td>51</td>
</tr>
<tr>
<td>3.6</td>
<td>Hubs and Authorities</td>
<td>52</td>
</tr>
<tr>
<td>3.7</td>
<td>Interpretation of SVD</td>
<td>56</td>
</tr>
<tr>
<td>3.8</td>
<td>Undirected bipartite graph</td>
<td>59</td>
</tr>
<tr>
<td>3.9</td>
<td>Hub and Authority graph respectively</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>Problems in HITS</td>
<td>67</td>
</tr>
<tr>
<td>4.2</td>
<td>Another problem in HITS</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Convergence graph for query “alcohol”</td>
<td>98</td>
</tr>
<tr>
<td>5.2</td>
<td>Convergence for query “affirmative action”</td>
<td>99</td>
</tr>
<tr>
<td>5.3</td>
<td>Convergence for query “death penalty”</td>
<td>100</td>
</tr>
<tr>
<td>5.4</td>
<td>Convergence graph for query “computational complexity”</td>
<td>101</td>
</tr>
<tr>
<td>5.5</td>
<td>Convergence graph for query “search engines”</td>
<td>103</td>
</tr>
<tr>
<td>5.6</td>
<td>Convergence graph for query “abortion”</td>
<td>103</td>
</tr>
<tr>
<td>5.7</td>
<td>Convergence graph for query “vintage cars”</td>
<td>104</td>
</tr>
<tr>
<td>5.8</td>
<td>The convergence graphs - Norm(2) algorithm.</td>
<td>105</td>
</tr>
<tr>
<td>5.9</td>
<td>The convergence graphs - Max algorithm.</td>
<td>106</td>
</tr>
<tr>
<td>5.10</td>
<td>The convergence graphs - SALSA algorithm.</td>
<td>107</td>
</tr>
<tr>
<td>5.11</td>
<td>Convergence graphs - Power Extrapolated HITS algorithm.</td>
<td>113</td>
</tr>
<tr>
<td>5.12</td>
<td>Convergence for query “affirmative action” - Hybrid Extrapolation</td>
<td>114</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

5.13 Two examples. .......................................................... 115

A.1 Convergence graphs for query “abortion” .......................... 136
A.2 Convergence graphs for query “affirmative action” .............. 137
A.3 Convergence graphs for query “alcohol” ......................... 138
A.4 Convergence graphs for query “amusement parks” ............. 139
A.5 Convergence graphs for query “architecture” .................... 140
A.6 Convergence graphs for query “armstrong” ..................... 141
A.7 Convergence graphs for query “automobile industries” ....... 142
A.8 Convergence graphs for query “basketball” ..................... 143
A.9 Convergence graphs for query “blues” .......................... 144
A.10 Convergence graphs for query “cheese” ....................... 145
A.11 Convergence graphs for query “classical guitar” ............... 146
A.12 Convergence graphs for query “complexity” ................... 147
A.13 Convergence graphs for query “computational complexity” .. 148
A.14 Convergence graphs for query “computational geometry” .. 149
A.15 Convergence graphs for query “death penalty” ............... 150
A.16 Convergence graphs for query “genetic” ...................... 151
A.17 Convergence graphs for query “geometry” .................... 152
A.18 Convergence graphs for query “globalization” ............... 153
A.19 Convergence graphs for query “gun control” .................. 154
A.20 Convergence graphs for query “iraq war” ..................... 155
A.21 Convergence graphs for query “jaguar” ....................... 156
A.22 Convergence graphs for query “jordan” ........................ 157
A.23 Convergence graphs for query “moon landing” ............... 158
A.24 Convergence graphs for query “movies” ....................... 159
A.25 Convergence graphs for query “national parks” ............. 160
A.26 Convergence graphs for query “net censorship” ............. 161
A.27 Convergence graphs for query “randomized algorithms” ... 162
A.28 Convergence graphs for query “recipes” ..................... 163
A.29 Convergence graphs for query “roswell” ..................... 164
A.30 Convergence graphs for query “search engines” ............. 165
A.31 Convergence graphs for query “shakespeare” ................. 166
A.32 Convergence graphs for query “table tennis” ............... 167
A.33 Convergence graphs for query “vintage cars” ............... 168
A.34 Convergence graphs for query “weather” .................... 169
List of Tables

5.1 Query Statistics ......................................................... 93
5.2 Users per query .......................................................... 96
5.3 Results of the experiments with Extrapolation ..................... 108
5.4 Top 15 results for query “search engines” .......................... 110
5.5 Top 15 personalized results for query “search engines” (α = 0.6) ........................................ 110
5.6 Top 15 personalized results for query “search engines” (α = 0.15) ........................................ 111
5.7 Top 15 results for query “computational complexity”, Exponentiated HITS .................................. 116
5.8 Top 15 results for query “computational complexity”, Personalized Exponentiated HITS (α = 0.05) .... 117
5.9 Top 15 results for query “affirmative action”, Randomized HITS .................................. 117
5.10 Top 15 results for query “affirmative action”, InDegree .................. 118
5.11 Top 15 results for query “affirmative action”, Personalized Randomized HITS (α = 0.85) ...... 118

B.1 Top 15 results for query “abortion” .................................. 172
B.2 Top 15 results for query “affirmative action” ....................... 173
B.3 Top 15 results for query “alcohol” ..................................... 174
B.4 Top 15 results for query “amusement parks” ......................... 175
B.5 Top 15 results for query “architecture” ............................... 176
B.6 Top 15 results for query “armstrong” .................................. 177
B.7 Top 15 results for query “automobile industries” .................... 178
B.8 Top 15 results for query “basketball” .................................. 179
B.9 Top 15 results for query “blues” ......................................... 180
B.10 Top 15 results for query “cheese” ..................................... 181
B.11 Top 15 results for query “classical guitar” .......................... 182
B.12 Top 15 results for query “complexity” .................................. 183
B.13 Top 15 results for query “computational complexity” ............... 184
B.14 Top 15 results for query “computational geometry” .................. 185
B.15 Top 15 results for query “death penalty” ............................ 186
B.16 Top 15 results for query “genetic” ..................................... 187
B.17 Top 15 results for query “geometry” .................................... 188
B.18 Top 15 results for query “globalization” ............................. 189
B.19 Top 15 results for query “gun control” ................................................................. 190
B.20 Top 15 results for query “iraq war” ................................................................. 191
B.21 Top 15 results for query “jaguar” ................................................................. 192
B.22 Top 15 results for query “jordan” ................................................................. 193
B.23 Top 15 results for query “moon landing” ......................................................... 194
B.24 Top 15 results for query “movies” ................................................................. 195
B.25 Top 15 results for query “national parks” ....................................................... 196
B.26 Top 15 results for query “net censorship” ...................................................... 197
B.27 Top 15 results for query “randomized algorithms” .......................................... 198
B.28 Top 15 results for query “recipes” ................................................................. 199
B.29 Top 15 results for query “roswell” ............................................................... 200
B.30 Top 15 results for query “search engines” ...................................................... 201
B.31 Top 15 results for query “shakespeare” ......................................................... 202
B.32 Top 15 results for query “table tennis” ......................................................... 203
B.33 Top 15 results for query “vintage cars” ......................................................... 204
B.34 Top 15 results for query “weather” ............................................................... 205
Chapter 1

Introduction

“Search is going to be the interaction paradigm”, John M. Lervik, CEO, Fast Search & Transfer [fas].

1.1 Introduction to Search Engines & Preliminaries

1.1.1 A Brief taxonomy of Search

Search engines today are totally different than maybe 5 years ago. Search is not just the search field; it is a platform and infrastructure. Search is equal to advertising, but advertising is driving just one side of the search. The other crucial side is the relevancy, match the user needs (informational or recreational), match people to people and connect the people.

The boom of World Wide Web (WWW) have made it easier to have access to various sources of information, reaching a wider audience than ever possible, and all kinds of digital communication provided greater access to networks. Information sources are available even if they are far away, they can be accessed quickly and effectively. Users can generously post whatever information that is in some ways considered useful by them. Users on the web are thus expanding from being an active information consumer to becoming an active information producer. With such a boom in information retrieval and information explosion, there is an ever increasing demand for access, quick responses, and relevant results. Thus there has been a much greater focus on a lot of different techniques that yield highly plausible results.

In this chapter we will broadly introduce the challenges and fundamentals of information retrieval (IR).

1.1.2 Search Engines and its Challenges

Search Engines are considered to be a vital part of the today’s fast growing era. According to the anticipation of key search engine companies everything now should come across the retrieval/filtering process – from information need to entertainment and foresightedly to any need. Everything must have to be searched or filtered first in order to have a better match to the needs. Varieties are growing rapidly, and to choose between those myriad varieties the seeker (user) needs to have an opportunity to materialize (specify) their requirements, which the system can recognize. The avalanches of varieties have baffled the seeker today.

Users of the information found themselves floundering, how to retrieve the relevant information. Today’s user formulates their needs and expects to have a match to their needs. That is, instead of finding it themselves
by brute force, they need a solution which can bring such facilitation. The users of the information systems hope to retrieve relevant information in an effective (best match) and efficient (fast) way.

We consider the user need as user query and information collection as the information. Most IR systems require user to formulate a query by transforming a complex representation of an information need into a short list of keywords. User query must therefore easily be formulated by the user, and efficiently recognized by the retrieval processes. There is an active research on the topic of user query formulation and language modelling, to better facilitate or make it easier for the users to formulate their queries [Met04].

The users want to save time by using an IR system, and expect to have a much better match of their requirements. The size of information is increasing at an incredible pace (exponentially, by many order of magnitude), with such an increase rate, the users are more concerned with the “time” it takes to retrieve a good match to their needs. Thus finding the required information from huge collections is an error-prone and frustrating task for the users. But today information retrieval or filtering is not just a facility – it’s an inevitable need, without retrieval by computational systems, retrieval is simply not possible. So, IR systems have become an essential part of today’s life.

The increase in the magnitude of information not only troubled the users but also the IR entities (such as, search engines, libraries, etc). It has been increasingly hard and challenging to satisfy the users’ needs from the huge and dynamic repositories of information. The explosive nature of information on the Web has further complicated the already difficult problem of identifying usable information from a large collection. There has been huge amount of ongoing research on characterizing and classifying information on these huge reservoirs, in order to make them more manageable for retrieval. Hence the intentions are to come up with some strategies or models that can bring the information closer to its user.

The information collections have been evolving in fascinating ways, apart from just getting larger and larger, it is also dynamically changing. The information sources are no more static, the results which were valid yesterday or maybe an hour ago or maybe a while ago are no more legitimate now. The entries added or updated in the information collections must be accommodated in existing ones. Keeping tracks of the updates and additions in the information collection and provide the users up-to-date information alone is a very gigantic target for research in IR. In addition, the users of IR also have a peculiar nature too – by having dynamically changing needs. So, the challenge is primarily to device a solution which can cope up with the dynamically growing information and dynamically changing needs of the users.

Personalized, context and content sensitive search

There is another interesting and active area in IR systems, i.e., to personalize the results according to user’s needs. The users of IR systems expect that their behavioural trends are monitored and based on that information personalize the search outcomes according to their preferences (see Chapter 4 for detailed discussion on personalization).

The results from the user query must also take into account the context of the user (from where in the globe the user is requesting for information), and the time at which the request is made. A user query “Christmas” should have different results in other months than in December, and should be different for users in different part of the world. Searches on the topics for which there are opposing communities such as, “death penalty”, “religion” or queries that are of interest to different communities, should be appropriately dealt with in IR. Thus the configuration and adaptation of results to different contexts should be possible. IR should be sensitive both to the content as well as context of retrieval.
1.1 INTRODUCTION TO SEARCH ENGINES & PRELIMINARIES

Storage Issues

Information repositories are proliferating incredibly and there is also a growing concern about the storage issues. Although today we can afford to have almost indefinite storage, but the price of operating on those indefinite reservoirs are startling. The challenge is how to represent and organize the huge collections of information, in order to retrieve and process them effectively. The huge collection of information entails a loss of performance and efficiency, because it takes time to process (index, cluster, etc), retrieve (query) and update information in the huge repositories. Thus there is a growing concern about the usability and the interaction time between user and IR systems. A trade-off between the quality of the results and query response time is mostly considered as an option. High quality and highly relevant results requires more effort in terms of computations hence a delay in the query time and vice versa.

The practical goal of IR systems primarily is to address many of the problems, both in quality and scalability, introduced by scaling search engines technology to such extraordinary numbers (huge information repository).

Relevancy

User query must yield meaningful, manageable and most importantly relevant set of results from IR systems. The results should be easily interpretable, and serve to have utmost relevance to what the user is seeking. And if there are many relevant documents to user query, they should be presented in an ordered manner, with the most relevant results appearing first. So, the users expect to have ranked results according to relevancy instead of unordered results.

It is commonplace for web search queries to have thousands or millions of results. The impatient user on Web doesn’t have the time and patience to go through all of them to find out the ones that they are interested in. Therefore it’s up to the IR system to provide only minimal set of results most relevant to the user query.

Relevance scoring and ranking is hence quite a crucial component of an IR system. The purpose of this study is also to explore different relevancy ranking models and their effects on overall search process. This study is primarily an investigation on improving or optimizing the existing relevancy ranking models or proposing any new concept or ideas that could be used to either speedup the interaction time or improve the quality of results.

Relevancy models range from traditional and simple approaches like Boolean based models, to vector based models (see Chapter 2), and more sophisticated approaches like link analysis based models (see Chapter 3). They all make extensive use of concepts from Boolean algebra, geometric representations, vector algebra, probability theory, and significantly the core concepts from Linear Algebra. The focus of this study is to travel from a general perspective of relevancy in IR to more specific and classy notion of “importance” in IR by doing Link Analysis Ranking (LAR).

Due to the importance of IR in general and large-scale search engines on the Web in particular a large amount of academic research has been done on them. Research range from classical and traditional retrieval models to the sophisticated and modern retrieval models using concepts from a wide range of disciplines.

Stability

In the link analysis based model, stability is a desirable property [Ng01b; Ng01a]. If an algorithm is not stable, then slight changes in the link structure of the network of documents may lead to large changes in the ranking produced by the algorithm. Usually users merely want to see the same results which they have seen before. If the algorithm is not stable enough, the results keep on changing from time to time.
Stability also provides some kind of “protection” against the malicious spammers. Because with stable ranking algorithms they will not be able to easily inflict the ranking by slight changes. Stability and locality therefore are very important properties of relevance ranking models, which enables them to prevent the widespread propagation of a change. Just like the perturbation analysis in linear algebra, stability and locality properties in relevance ranking models concerns with the consequences of changes on the search outcomes.

1.1.3 Elements of search

Documents

Documents have quite a central role in IR process. They either contain the right information or the clue of where the right information can be found, in the form of citations.

Document can be a text file, an html file, a pdf book or article, a media file, a blog entry, an image or any other type of object representing information. In short document is a searchable entity in IR and is conceptually more or less equivalent to database table or entries in the table. Documents must be stored and processed in a manner which provides a better result and swift throughput overall. It has become increasingly promising to have more queriable documents, with the actual elements of the document being tagged, e.g., like XML. Structured or tagged documents (chapters, sections, subsections etc) with optimized representations in index structures directly affect the overall performance of IR.

How IR system views the document is also very crucial, for example, in order to logically or conceptually view the documents, text operations such as stemming and lemmatization play a vital role. The logical view of the document might be considered as a shift smoothly from a full text representation to a higher level representation such as concepts or ideas deduced from the document.

Document structure

Storing information regarding internal structure of the document such as, chapters, sections, subsections etc, helps to provide a much precise understanding of documents and a much defined way of presenting them to users.

There is yet another very important piece of information about the structure of documents, i.e., their citation structure. The citation structure of documents provides a valuable source through which the importance of document can be evaluated from its neighbourhood in terms of citations. If a lot of other documents cite one document that means the document in question is important in some particular topic(s). Link analysis ranking models exploits the citation structure of the document to provide popular and important set of documents, relevant to user query or important in general independent of query.

It’s quite beneficial and depending on algorithm quite essential to exploit the document’s internal structure to provide a more focused and controlled retrieval. The aspects of the DBMS (the fields) in the structured IR, and the schemes of XML retrieval could be exploited in structured document retrieval. There are also places where we would like to shift from structured to non-structured documents and vice versa. Mapping from/to structured to non-structured document is important in such situations.

Natural Language Analysis

The Natural Language Analysis techniques can be used to better understand the documents in the corpus, in order to provide a good match between user query and documents. The user formulates their queries in natural language formalism and expects from IR system to understand their needs and bring forth the information
relevant to those needs from the collection. Natural Language Processing (NLP) is a discipline in itself, within Artificial Intelligence, but some of the techniques from that discipline could be of utmost relevance when it comes to content-analysis and text operations in IR.

**Meta Information**

Meta information is the information that can be inferred about a document, but is not contained within the document. Examples of external Meta information include things like reputation of the source, update frequency, quality, popularity or usage, and citations. Meta information have widespread applications, for example, in LAR the excessive exploitation of popularity and citations meta information, and in personalization the application of ‘usage’ meta information.

**Query**

Formulating a “good” query is still a black art. In particular, many inexperienced and layman users do not have the right skills to formulate a sufficiently precise query which a system can recognize accurately. Most of the times instead of writing query keywords; they ask questions, e.g., ‘what is information retrieval?’.

Query subsystem is the other main element of an IR system, which also constitute a considerable influence on the overall search. A misrepresented query will always end up in topic drift\(^1\). The closer the representation of query to the representation of documents the better will be the relevancy of results to the query.

Shifting the burden of enriching their needs in form of a good query that an IR system can understand, the user expects from query subsystem to predict their needs from their sloppy queries. Their expectation is driven by their unawareness of the internal representation of documents in the IR system. It is mostly the query subsystem’s responsibility to bridge the representation of query with the internal representation of documents by enriching the user queries with required information needed to decipher them.

The query subsystem takes care of enriching the user query with the required structure and then sends that enriched query to the internal search processes for further processing. Query processing therefore entails, configuring the queries, natural language processing, obtaining personalization information, and thereby supplements the queries with required information.

**Users**

Users of IR system are responsible foremost to appropriately specify their needs in form of a “good” query. Once the query is submitted, the users may also be required to provide relevance feedback on the search outcomes, which could be achieved in cycles. The explicit user relevance feedback is employed to purify the search outcomes and provide a better match to the query, and consequently keep users’ behavioural information for personalization. The implicit user behavioural information can be acquired from their usage logs on webservers. The interpretation and utilization of implicit user feedback is a contemporary mechanism for obtaining users behavioural trends, in order to improve the ordering of top results in search.

**1.1.4 World Wide Web (WWW)**

WWW is networks of recommendations and it can be related to the citation structure of documents. The citation (network) structure of the WWW (the hyperlinked environment) provides a rich source of information, provided

\(^1\)When the documents returned by the IR system is not relevant to the user query, than we say that there is topic drift, i.e., the document does not correspond to the query topic.
we have effective means to understand them. Considerable benefits can be accomplished merely through a pure
analysis of the link (citation) structure. By analyses of the link structure we tend to find the classification of
the documents by figuring out from the link structure what the author or creator of documents intended to
classify (implicitly or explicitly). These analyses will help us identify the proximity and relevance of documents
amongst each other. And enables us to find out the social or informational organization of the documents.

The work done in LAR models are primarily the successor to the Bibliometrics, study of written documents
and their citation structure [Lar96]. And it is essentially focused to purely exploit the citation or link structure
of the documents.

Hypertext Information (Network structure)

Hypertext information encodes a considerable amount of latent human judgment which can be used to formulate
the notion of importance from graph theoretic perspective. The hyperlinks therefore define the “context” in
which a document appears. We utilize the contextual exposition of documents to deduce the importance or
popularity of the documents in the network, by using the core graph theory concepts and techniques.

Web Graph (network)

In the evaluations and analyses of the algorithms in the network or hyperlinked based models, it is fundamentally
important to form a legitimate classification and understanding of the graph (network of documents). Through
either clustering techniques, or applying some other scheme on the adjacency matrix of the graph we could
achieve a better classification of the network of the documents. These classifications enable to enhance the
visualization of the webgraph by grouping together similar documents. Thus it allows us to inspect how the
top-ranked results of the different algorithms are interconnected with each other and with the rest of the graph.
This kind of analyses yields significant insight into behaviour of the algorithms by visually depicting their
expositions to the connectivity in the network structure.

Clustering techniques thus involve dissecting a heterogeneous population of documents into subpopulations
that are in some way more cohesive or maybe homogeneous [Dri99].

Web search and traditional IR

Web search is peculiar in nature in contrast to the traditional IR models. The document corpus is much more
dynamic in nature and much more diverse in posture; ranging from structured to non-structured and from media
to blogs and many more. The document corpus in web search is a hypertext corpus of enormous complexity,
and continues to grow in size and density at a phenomenal rate. Scalability therefore is much more important
in web search than in conventional IR. How much scalable is the search in comparison to the growing size of the
information, is quite a central question in web search. Thus in web search the growing size of the documents
collection should be smoothly and transparently dealt with, in order to offer persistent outcomes.

The needs and behaviour of web users and traditional IR users are also different. The web user is hasty in
nature, mostly not as serious as in the case of traditional IR.

Notion of similarity in traditional IR perspective means to match the query strings to the documents’
strings. Thus relevancy in traditional IR implies that the query strings or query concepts is relevant to the
document strings or concepts. While in web search the notion of similarity is largely irrespective to text but
subject to the context where documents exist. In essence in web search the focus shifts from “relevance” to
“authoritativeness”. If the documents are popular or authoritative in the query topic in their own contexts, then those documents are deemed to be important and relevant.

Specifically, in the web search there is more focus on the link or citation structure of documents and therefore more focus on behaviour of the network of documents. Analyses of the network structure and its properties thus provide much more farsighted benefits than just pure text analysis. Most of the new models in LAR use the core graph theory concepts like \textit{in-degree}, \textit{out-degree} and \textit{reach-ability} of the network(s) of documents to formulate the notion of similarity.

Ranking purely by in-degree (popularity) can benefit in several cases but have problems also in some other cases. There might be documents with large number of in-degree but lack any thematic unity. Therefore applying different simple heuristics might sufficiently help to supplement the ranking based on in-degree. There has been studies in the area where text based techniques (content-analysis) and probabilistic model for modeling the contents are used in conjunction with the link (connectivity) analysis [Bha98; Coh01; Met04; Kai98]. The proposition of basic heuristics by which hyperlinks can enhance the notions of relevance of a document is quite diverse and central to the area of link structure analyses.

\textbf{Convergence}

One of the concerns in the link based ranking methods is the convergence to a “good solution” or an equilibrium state. A good solution or an equilibrium state is a state where system under certain presumptions can declare the set of good results corresponding to user query. Most of the link analysis based ranking models are iterative in nature, they iteratively move towards the required equilibrium state (the good solution). One of the contributions of this study is the speedup of the convergences of the famous LAR algorithms (see Chapter 4).

Convergence is a central phenomenon in iterative algorithms [Lay94]. In linear algebra, the iterative methods are employed when direct methods would be prohibitively expensive and in some cases \textit{impossible} even with best possible computing power to find out the actual solution. Essentially the iterative methods provide an approximation to the true solution.

\textbf{Evaluation}

For the evaluation of the relevancy models in link based models first we need to have techniques for producing “enriched” samples of the document collection to determine notions of structure and quality that make sense globally. Evaluation of the link analysis based models is therefore a difficult task. The difficulty is primarily due to unavailability of a standard framework for evaluation and a representative set of data, which could be confidently used to assess the quality of the algorithms. Tsaparas [Tsa04a], has presented a limited framework for comparing and analyzing LAR algorithms.

A true test of the quality of a search engine would involve an extensive user study and results analyses. The authenticity of outcomes presented as a result of search is quite important, both for user and as a general ethics in IR. For example search for president name should give first the serious and authentic information than any joke or bizarre information. In results analyses, authenticity of the search results must also be evaluated. In our evaluation we would mainly rely on the user study done in [Tsa04a].
1.2 Contributions

As a result of this study we have made the following contributions which in some ways address some of the challenges broadly discussed in this chapter.

- An extensive exploration of the general concepts and ideas in IR to a certain extent. Several fascinating ideas have been studied which consequently stimulated the next step of the thesis. From the wider perspective of IR got a purposeful motivation towards LAR. LAR models are then explored to certain profundity, in order to form a strong basis for further and deeper analyses.

- In the evaluations and analyses we have incorporated a very effective and novel technique of Extrapolation in query-dependent LAR algorithms. This technique offered a very radical improvement in convergences of the pioneer algorithms such as, HITS and SALSA. Hence application of Extrapolation to query-dependent LAR algorithms is the major contribution of this study, primarily because this idea was not yet explored in LAR research.

- Studied personalization and formulated some of the query-dependent algorithms within framework of the generic personalization model.

- Extensive experimentation of the ideas explored. Both Extrapolation and Personalization are exposed to empirical evaluations to see their behavioural exposition in practical settings.

One of the major contributions of this study is also the exploration of wide range of ideas and the extensive experimental evaluation of a relatively small subset of those ideas. The study is novel because it contains both theoretical glories and empirical evidences.

In rest of the thesis I have elaborated on the aforementioned contributions. Henceforth, the sketch of contents for rest of thesis is portrayed in the next section.

1.3 Outline

The remainder of the thesis is devoted to look into the challenges that we have discussed quite broadly in this chapter, and also in line with the objectives of the study. To address the objectives I have divided the thesis into three main parts.

Following is the brief outline of what we will learn in the rest of the chapters:

- In Chapter 2 we discuss IR models in general, present classical retrieval models and their respective roles in motivating the more sophisticated and newer models. We also review the related literature on IR models in the same chapter.

- In Chapter 3 we introduce the concepts and elements in LAR from literature, the existing models in LAR, their contribution to help resolve the search goals. Both Chapter 2 and 3 form the necessary theoretical background to form a firm basis for the forthcoming analyses and evaluations. Therefore both Chapter 2 and 3 are broadly considered as first part of the thesis, the theoretical background part.

- In Chapter 4 we analyze further in depth the models introduced in Chapter 3, specifically with a focus on query-dependent LAR models. We elaborate on the important properties of the algorithms and their intrinsic limitations, and subsequently put forward appropriate adjustments which can possibly eliminate
or lessen their side-effects. We also unveil the necessary mathematical implications of the algorithms. Later advocate the novel idea of extrapolation to speedup the rate of convergence. Hence we evaluate the core properties of LAR algorithms such as, their convergences, stability, uniqueness and existence of good solution(s). In the end of the chapter we also describe different approaches towards personalization, and in the light of theory formulate a generic framework for personalization. In the same section we focus on personalizing the query-dependent algorithms, and explore and exploit numerous ideas for personalization.

- In Chapter 5 we assess the experimental implications of the algorithms and concepts discussed in Chapter 4. The focus of this chapter is therefore to see the relation between the theoretical assertions and practical outcomes. We also present the comparative evaluation of different LAR algorithms, in form of tables and graphs. We have presented all the results of the experimentation in appendices (Appendix A and B). Chapter 4 and 5 constitute the second part of the thesis, the evaluations, analyses and experiments part.

- Chapter 6 concludes the thesis with the important results and possible future work, to extend or carry on with the various areas discussed in this study. We also compare the objectives of the thesis with the overall outcomes of the study. Contributions of this study are discussed with more details together with their much broader implications. Ultimately this chapter winds up with the final part of the thesis, the conclusions, recommendations and future work.
Part I

Theoretical Background
Chapter 2

Information Retrieval Models and trends

2.1 Information Retrieval (IR) Problem

The IR problem can be broadly viewed from two main perspectives, one as a computer-centred and other as a human-centred.

The computer-centred view of IR problem is essentially concerned with issues like improved crawling and indexing approach, highly efficient ranking algorithms, time efficient replies, efficient representation of the documents, and highly relevant result sets. The utilization of structural information of documents (chapters, sections, subsections, etc) and their representational capabilities is yet another fruitful area in computer-centred approach of IR. This useful source of information provides necessary locality information within the documents, which could be used to assess the relevance of document to a query topic. How the structural information is used; can an IR model function on both the structured and un-structured documents; these are some of the interesting problem statements in this area. There has been a rich amount of research done in only the computer-centred view of IR problem, because of its complexity and dynamics [He05] (see also later sections, and Chapter 3).

The human-centred approach on the other hand is primarily concerned with the behaviour, needs and requirements, representations, enrichment and disambiguation of the user queries. The main focus therefore is to understand the users’ need, and determine how the involvement of users can benefit the organization and operations of the IR system. The purpose of the human-centred approach is to involve the user more actively in the IR process. For example allowing user first to write their queries in the natural language formalism, the system then fetches the results on this initial query, the user goes through the results and filter out irrelevant results, the system refines the query further on users feedback and then provide much better outcomes. The understanding and implementation of implicit and explicit user relevance feedback can be used in large scale operational environment to improve retrieval. Implicit and explicit user feedback in human-centred approach has been thoroughly studied independently and together with other relevant approaches by different research groups [Agi06a; Bau01; Cha00].

There has been a much greater tendency observed towards human-centred approach of IR problem now [Tan02; Jeh03] (also see Chapter 4). The efforts towards collecting more user relevance feedbacks are primarily destined to purify and personalize the search outcomes. By doing that, it is expected to have improved retrieval and hypothetically bridge the user needs for information with the systems’ internal representation of information.

In this chapter our focus is on the IR problem both from computer and human-centred perspective of IR.
We will bring up the candidate point of views in terms of methods and approaches used to solve and deal with the IR problems identified so far. First we define the IR process broadly and then in the later sections elucidate about different IR models with an objective to explore the relevant concepts to the focus of our study.

2.2 The IR Process

The IR process can also be considered from two main viewpoints, the *query-independent* view and the *query-dependent*. Figure 2.1 presents the two viewpoints in graphical form. We will explain the objects and concepts in this figure during the proceedings of this section.

The IR process intrinsically is quite wide and complicated; describing it generally and sufficiently will just serve as a very top level overview. In this section we will therefore have a broad impression of the IR processes, which could be used to establish a basis for realizing the concepts and approaches used in IR later in this study.

2.2.1 Document Corpus

Before anything else in the IR process, there should be a *document corpus* or the document database (e.g., a database of articles, books or webpages). This is the place from where the *contents* or documents get into the retrieval databases. The feeding of the corpuses and collections with the required content is done here. Content feeding is therefore an ongoing process in the life cycle of retrieval, the contents keep on adding, updating and removing. And these are the task-lists of the content feeder, as shown in figure 2.1.

In the case of the World Wide Web (WWW), the *crawler subsystem* collects and stores the documents from all over the web and feed it to the document corpus (permanent storage). In case of traditional IR systems, such as libraries or electronic archives of articles or other materials, the document databases could be populated and feded in different ways (e.g., manually storing the pdf version of documents or any other automated mechanisms).

2.2.2 Document Manager

After having a repository of the documents the *document manager subsystem* will do the necessary operations. The document manager enforces structure to the documents by applying a set of text operations on them. It depends on overall search paradigm, how it sees the documents and what kinds of information are deemed important for retrieval. The text operations broadly transform the original documents by generating a *logical view* of the documents. The logical views of the document are actually the interaction points with the original documents. The logical view contains sufficient information about the document (contents, concepts, structures, etc), which could be used in the later steps in IR process. Efficient and effective logical view (a good representation of the original document) gives a direct boost to the retrieval both in terms of performance and relevance. The logical view of document is formed by applying the natural language operations like, stemming, lemmatization, elimination of stopwords, lexical analysis, recognition of the structure (chapter, section), and other important operations which can help to represent the document accurately. These text operations (transformations) are meant to provide a representative and thematic overview of the documents, depicting the concepts, ideas and structures available in the document. They are primarily used to cope with different IR problems like *Synonymy* and *Polysemy*, words having same meanings or words referring to the same
Figure 2.1: Information Retrieval Process
concepts respectively. But their main advantage is to provide a representation of the documents which could be effectively used on behalf of the original document, for retrieval processes like matching and ranking.

### 2.2.3 Indexer

After identifying the logical view of the documents which is done by the document manager, the indexer subsystem will take the charge. Based on the logical view, indices are created, which help to efficiently traverse and search for the relevant documents. An index structure in IR model is conceptually similar to the indices in books, which helps to search through huge amount of data swiftly. The primary purpose of the indexing is to provide a data structure which can ease the retrieval tasks in IR. Indexing is a wide area in itself in the IR systems. If indexing is not properly taken care of, it incurs considerable performance loss. The data structure produced by the indexing module is used at query time, which means indexing has a direct impact on the search efficiency.

There are a lot of indexing structures available, for example, the most famous ones are inverted files, content index, structure index and other special purposes indexes (image and pdf index, etc). Other valuable information like the hyperlink structure of the documents can also be kept as an index structure, which will help the ranking subsystem to make use of it. Most IR systems employ numerous index structures at a time depending on the type of the documents, the kind of information available in those documents and essentially on the user/system requirements. The multiple indices are employed either in parallel or synchronously during the matching process.

The need for efficiency in index structures mainly depends on the amount of query needed to handle per second (or milliseconds), and the storage required to keep those indices. One of primary considerations when introducing a new index structure is the space it occupies. The document databases are usually enormous, and therefore, the index structures corresponding to those enormous databases also occupy huge amount of storage unless properly dealt with. Therefore compression, decompression and filtering techniques have been increasingly used as an integral part of the indexing.

Once the indexing is done the query-independent part of the IR process finishes (see figure 2.1), and now the query-dependent (interactive) part could be commenced. The information seeker from the IR interface will specify the need for information in form of a query; a set of keywords or regular expression with wildcards. The formulation of the user query and its interpretation depends on IR model’s specification. After advent of World Wide Web, there has been a consistent effort applied to come up with models which can help ease the users in formulating their queries effectively [Bau01; Met04].

### 2.2.4 Query processing

The user queries after submission are parsed and then sent to the query processing subsystem to enrich them with the necessary and system specific details. The text operations during query processing on the user query will also create a logical view of the user query. The text operations are the same natural language operations as in the case of the documents. After the logical view of the query is generated, the system specific representations of the user query will be formulated. The resultant enriched query can be recognized and subsequently processed by the later steps in retrieval.
2.2.5 Matcher

Once the system representation of the query is generated after query operations, the improved query is sent to the core search process, the matcher subsystem (see left side of the figure 2.1). The actual matching of query and documents will take place here in order to fetch the set of relevant documents. The matching will be based either on concept matching, or keyword matching, or other matching techniques depending on the IR model’s strategies. At this point different IR systems use different heuristics, algorithms and strategies to better predict a match between user query and the documents. As discussed above, depending on how good the indexing is done, the matcher subsystem will perform accordingly. Efficient and comprehensive index structure will yield faster and effective matching between queries and documents.

2.2.6 Ranking

The retrieved documents set from matcher subsystem are usually quite large; presenting them as it is would not be useful for the user. So, the retrieved documents must go through the ranking subsystem, to be ranked according to the likelihood of relevance to user query. The output of the ranking is an ordered list of results such that the documents near the top of the list are most likely relevant to user needs.

Ranking is usually done at the query-dependent part of the IR process, but there are IR models which calculate part of the ranking scores independent of query (e.g., at crawling time). Some of the models in Link Analysis Ranking\(^1\) compute ranking scores independent of any query in order to reduce the overhead in the Ranking subsystem (see Chapter 3 for further details).

The purpose of ranking is to make it manageable for the user to interpret the results from the search. The relevance can be determined on many factors, such as the users’ personalization factor, which can be the users’ contextual information and users’ historical information (browsing history, usage logs, bookmarks, etc). There are many other dynamics independent of users, such as; the frequency of the query terms in the documents (\(tf-idf\) factor), where in the documents (heading, abstract, body) user query terms are found, the popularity of the documents in their respective contexts. The different ranking strategies or algorithms will be discussed in later sections.

2.2.7 User Feedback

The retrieved and ranked documents are ready to be sent back to the users, who will tend to examine the relevance to their needs. At this point a user feedback loop could take place depending on IR model. The user feedback cycle is meant to better understand the users’ needs, provide them a more relevant results, and/or personalize the results according to their predefined interests. The user feedback could be either explicitly gathered (by feedback loops) or implicitly obtained through their interaction with the search results (e.g., click-through, time spent on document, document dwell time, etc) [Agi06a; Agi06b].

From the above scenario it seems quite straightforward that users’ information needs will be satisfied intuitively. But the reality is in contrary, there has always been petulant users seen, who do not get satisfactory results from querying the IR system. Since most of the users are unaware of the text and query operations, the queries that they usually formulate are mostly insufficient (lacking sufficient expressiveness). And therefore they usually get irrelevant results, because of the fact that they have not formulated their queries appropriately. Hence poorly formulated queries results in unsatisfactory and irrelevant result set, and therefore annoyed users.

\(^1\)PageRank algorithm calculates ranking at a time independent of query.
There are a lot of other factors involved in an efficient IR model; we will go through quite a few of them in next sections, and try to relate them with the focus of this study. Starting with the overview of the traditional models in IR in the next section.

2.3 An Overview of the Conventional IR Systems

One of the central components of the IR system is to rank the retrieved documents, as identified earlier. Ranking illustrates the relevancy of the retrieved documents set to the users query. The issue of predicting which document is relevant and which is not relevant, is quite a central matter in IR systems. It’s not far from truth to consider ranking as core of the IR systems. The notion of relevance has been widely explored in order to device efficient algorithms which provide more relevant results to the users. Different IR models determine relevancy in their own peculiar ways, so the question of what is relevant and what is not, depends primarily on the IR model under consideration. There have been a lot of different models proposed over the years. We will discuss a few of the relevant ones here, starting from the three classical and primitive models:

- Boolean model
- Vector model
- Probabilistic model

The implicit assumptions in the classical IR models are first that document corpus is fixed and well structured and manageable in size. Secondly the users are assumed to be trained and cooperative, thus the overall environment in the conventional IR models is much more controlled in comparison to contemporary IR models. Thirdly the initial goals of the classical IR models were not that complex, just to fetch documents relevant to query and ignore context, user specific information and other complex representations. Here we start with these simple classical models and go into much complex models later in the study.

2.3.1 Boolean model

Boolean model is one of the primitive and simplistic IR models, which uses the principle of exact matching, in order to match the documents with the user queries. The Boolean IR system require the users to specify their queries using a complex combination of the Boolean logic operators, ANDs, ORs and NOTs. The model is also thought to be a set theoretic in nature. A more refined version of Boolean IR model is still in use for example, in electronic libraries. “Boolean” is named because the terms or keywords in the user queries are joined and compared with the documents using the Boolean algebra. All the terms in the queries are used as it appears in the query, and matched with the documents, and results are presented without distinction or enumeration.

The user query ‘X AND Y’ means to fetch all the documents which contains both query terms X and Y. So, the formulation of the query is dependent on the capability of the users to transform their needs into Boolean expression. Considering the fact that any logical statements can be expressed using the Boolean logic, it apparently seems quite promising. The queries are specified as Boolean expressions which have a precise semantics. But the users find it difficult and awkward to express their queries in terms of Boolean expressions.

The query operations in the Boolean model are simplistic usually done by means of regular expressions. Matching only involves to find presence or absence of keywords in the document and relevancies are judged on scale of relevant or irrelevant, there is no concept of partially relevant. This is one of the main disadvantages of this approach that such type of exact matching produce either too few or too many documents.
2.3 AN OVERVIEW OF THE CONVENTIONAL IR SYSTEMS

But the variants of the Boolean model have been introduced which exploit the inherent advantages of this primitive model and get rid of the inherent limitations, by replacing them with any other germane ideas. For example, the Fuzzy Boolean model makes use of the fuzzy logic to incorporate the idea of partial matching in Boolean based models. We have discussed a few of the alternative Boolean based models in the later sections.

The Boolean model also fall prey of the two conventional problems in IR, the synonymy and polysemy. Synonymy refers to words having same meanings such as ‘car’ and ‘automobile’, while polysemy refers to words having multiple meanings such as ‘bank’ (could mean bank of a river or financial centre). These problems together with the intrinsic inabilities of Boolean model may result in a lot of irrelevant outcomes.

The Boolean model also requires the user to have the knowledge of syntax of the query. A user who forgot to put the quotation mark around a phrase might get a lot of irrelevant results.

Boolean model essentially does not allow any kind of relevance ranking of the retrieved document set, although some of the documents are more relevant to the user query than others. Excluding documents which do not precisely match the query terms results a major drawback and ineffectiveness of Boolean model. Thus, Boolean models in reality is much more of data (instead of information) retrieval. And therefore Boolean model is considered weak and ineffective for its general use in IR.

But the variants of the Boolean model do perform equally well. The reasons are that they exploit some of the core advantages of the Boolean model. Creating and programming Boolean model is straightforward and simplistic. Secondly and importantly, queries can be processed quickly, because each query term can be searched in parallel and joined according to the operator used. Thirdly the Boolean model scales well to very large document collections, hence accommodating the growing size of collection is much easier [BY99].

Despite the inherent disadvantages of the Boolean model many devotee users still use Boolean systems as they feel more control of the retrieval process [Sin01]. It is also still in use in the traditional data retrieval systems, such as database management systems.

2.3.2 Vector model

The vector based model has identified the inherent shortcoming of Boolean based models that the use of the binary weights is too restrictive, and proposed a solution in which partial matching is objectively possible. Developed by Gerard Salton in the early 1960s, to sidestep some of the problems identified in Boolean based models [Lan06].

Instead of using binary weighting (relevant or irrelevant, 1 or 0) the vector model uses non-binary weights to index terms in the queries and documents. Hence both documents and queries are represented as vectors in terms space. By doing this, it takes into consideration the documents which are partially similar to the query terms. And therefore calculates the degree of similarity between the documents and the query terms. The documents are ranked in decreasing order of degree of similarity. Thus vector based models return documents in an ordered list, sorted according to the relevance to the user query. The result sets are a lot more usable than result sets retrieved by Boolean model [BY99].

Formulation

In essence in vector based model the textual data in the query and document are transformed into vectors, and then employs vector algebra techniques to discover the key features and connections between the documents and queries. Therefore a document \( d_j \), and a query \( q \) are represented as \( n \)-dimensional vectors, as shown in figure 2.2. The proposition of the vector based model is to evaluate the degree of similarity between the
Figure 2.2: Vector model

document $d_j$ and the query $q$, by correlating them to the vector $\vec{d}_j$ and $\vec{q}$ respectively. This correlation can be quantified, for instance, by cosine of the angle between the query and document vectors, i.e.,

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|}$$

(2.1)

where $|d_j|$ and $|q|$ are norm (such as Euclidean norm) of the document and query vectors. A particular query vector $\vec{q}$ is constant among all the documents, its norm will be same for all the documents, while the norm of document provides a normalization in the space of document.

Typically, the angle between the two vectors (see figure 2.2) is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity (since cosine has the nice property that it is +1 for identical vectors, i.e., $\cos(180^0) = 1$ and 0 for orthogonal vectors, $\cos(90^0) = 0$).

Since the $sim(d_j, q)$ varies from 0 to +1 (inclusive), the vector model therefore ranks the documents according to their degree of similarity to the query vector. Depending on the threshold set for the $sim(d_j, q)$, the documents can be retrieved even they are partially relevant to the query (having value greater than threshold between $[0 - 1]$ ).

Clustering

Index term weighting can be calculated differently, depending on requirements and specifics of the model. There have been greater concerns on the ideas to figure out the most effective term-weighting techniques for vector based models in specific and generally for wider use in IR.

One of the proposition is for example to support clustering techniques [Dri99]. From the entire document collection, the goal of clustering primarily is to separate the document collection into two sets: documents which are related to a set $A$ (a vague set), and documents which are not related to the set $A$. While more sophisticated clustering might attempt to separate the document collection into various collections, for example clustering of
documents that are syntactically similar [Bro97]. However the decision to assign the query to different clusters is imprecise.

In the scenario of the clustering, the IR problem is reduced to the problem of determining which documents are in the set $A$ and which are not. This means viewing the IR problem as that of clustering problem. But there are two main issues in the clustering problems: *intra-cluster similarity* and *inter-cluster dissimilarity*. Intra-cluster similarity provides quantification of the need to determine, what are the features which better describe the objects in set $A$. While in inter-cluster dissimilarity one needs to determine what are the features which better distinguish the objects in the set $A$ from the remaining objects in the collection. The most successful clustering algorithms tries to do a balance between these two issues, intra-cluster similarity and inter-cluster dissimilarity [BY99]. Clustering is one of interesting areas in IR; there have been wider and prospective implications of this concept in IR. Clustering is a well-studied subject and there are many notions of how to measure the effectiveness of a cluster. Some of interesting contributions are found in [Dri99; Bro97].

Here in vector model we have a much specific and limited implication of clustering, described in the following subsection.

**TF and IDF**

*Term frequency, $tf$,* the number of occurrences of the query term in the document, is widely used as a weighting measure for documents ranking. In the perspective of vector model, the term frequency factor $tf$, corresponds to the intra-cluster similarity, and provides a measure of how well the terms describe the document contents. While *inverse document frequency factor, $idf$,* corresponds to the inter-cluster dissimilarity. The motivation for usage of idf factor is that terms which appear in many documents are not very useful for distinguishing a relevant document from the non-relevant ones. However the term frequency $tf$ is dependent on the size of document, i.e., the number of terms present in the document. Thus there is a chance that this measure for term weighting can be misused. As a consequence, $tf$ needs to be normalized by using techniques like term frequency normalization [He05].

For the query term weights, Salton and Buckley [BY99] suggest:

$$w_{i,q} = \left( 0.5 + \frac{0.5 \text{freq}_{i,q}}{\text{max}_i \text{freq}_{i,q}} \right) \times \log \frac{N}{n_i} \quad (2.2)$$

where $\text{freq}_{i,q}$ is the raw frequency of the term $k_i$ in the text of information request $q$.

There are now several variation of the term weighting identified initially by Salton and Buckley [BY99; Rob04; He05].

**Advantages and Disadvantages**

Vector based model broadened the perspective of IR models after Boolean based models. There are many advantages of the vector based models such as; the term-weighting schemes gives an edge to the retrieval process; the partial matching allows the documents which are partly relevant to the query to appear in the relevant document set; the cosine ranking sort them according to the degree of similarity and therefore provide an easily manageable list to the user.

The disadvantage of the vector based model is primarily its computational expenses. At query time similarity measures must be computed between each document in the collection and the query. Due to the exponential growth of information collection, the computation becomes further prohibitive. Apparently the vector based model doesn’t seem to scale very well to the increasing amount of the information.
The other potential disadvantage of the vector based model theoretically is that the index terms are assumed to be mutually independent. And due to the locality of many term dependencies, their arbitrary application to all the documents in the collection might in fact hurt the overall performance.

Despite all these problems, vector based model is widely used and considered to provide a resilient ranking strategy [Jon]. Mainly because vector based model is hospitable to other theories. There has been extensive comparison of vector based models with a large variety of other ranking strategies, but the consensus seems to be that, vector based model is either superior or almost as good as the known alternatives [BY99; Sin01].

2.3.3 Probabilistic model

Probabilistic model is based on the probability ranking principle, i.e., it tends to rank the documents in the order of their probability of relevance to the query, given that sufficient evidences are available. The probabilistic approach was first presented by Maron and Kuhns (1960). The model was later introduced in 1976 by Robertson and Sparck Jones, which afterwards became known as Binary Independence Retrieval (BIR) model [Jon]. The broad intention behind the probabilistic model is to capture or envision IR problem within the framework of theory of probabilities.

Ideal result set

Given a user query, we define an ideal set as a set of results which contains only relevant results and no other. Given the properties of the ideal result set we won’t have problems to retrieve documents comprising the ideal set. Querying can be thought of as the process of specifying the properties of the ideal result set (which can also be thought of as a clustering problem). The only problem with this newly created IR model is that we do not really know the properties of the ideal result set. We only know the index terms, whose semantics should be used in some ways in order to characterize those properties. Since the properties are not known, an initial guess has to be made, as what these properties could be. These initial guesses will serve as the preliminary probabilistic description of what the ideal result set will be, and they will be used to retrieve the initial set of documents. And then as the probabilistic model recursively operates, it requires that the user take a look at the retrieved documents and decide which ones are relevant and which ones are not, the system successively tend to improve the description of ideal result set. By repeating this iterative cycle many times, it is expected that a suitably well description of the ideal result set can be evolved which will be closer to the original description of the ideal result set.

Formulation

Given a query \( q \) and document \( d_j \), the probabilistic model tries to estimate the probability that the user will find the document \( d_j \) interesting and relevant. As explained above the model assumes that there is a subset of the documents which the user prefers as the ideal result set for the query \( q \). The assumptions are a bit vague because they do not state explicitly how to calculate the probabilities of relevance.

Retrieved documents are ranked by their odds of relevance (the ratio of the probability that the document is relevant to the query divided by the probability that the document is not relevant to the query). Taking the odds of relevance minimizes the probability of an erroneous judgment [BY99; Jon].

\[
\text{sim}(d_j, q) = \frac{P(R|\vec{d}_j)}{P(\overline{R}|\vec{d}_j)}
\]
The index term weight variables are all binary. A query $q$ is subset of the index terms. Let $R$ be the set of documents initially known (guessed) to be relevant, and $\overline{R}$ is complement of $R$. Let $P(R|d_j)$ be the probability that document $d_j$ is relevant to the query $q$ and $P(\overline{R}|d_j)$ be the probability that $d_j$ is non-relevant to $q$.

Using Bayes’ rule [Lay94],

$$sim(d_j, q) = \frac{P(d_j|R) \times P(R)}{P(d_j|\overline{R}) \times P(\overline{R})} \quad (2.4)$$

where $P(d_j|R)$ stands for probability of randomly selecting the document $d_j$ from the set $R$ of relevant documents.

Advantages and Disadvantages

The advantage of the probabilistic model in theory is that documents are ranked in decreasing order of their probabilities of being relevant. Secondly, it poses new areas and therefore a family of the IR models (probabilistic or logical retrieval models) continued to grow. One class of probabilistic models which has been used extensively in other application areas consist of those based on networks, such as inference networks [Jon] (see Section 2.7.2).

Unfortunately it can be very hard to implement the probabilistic model. Their complexity grows quickly, deterring many researchers and limiting their scalability. Because the initial guesses which need to be made about the ideal result set is vague and can be very complex in the case if document collection is enormous and heterogeneous. The method also does not take into account the term frequencies ($tf$) with which an index term occurs inside a document. The assumption of independence of the index terms as well as documents might be prohibitive in practice (see Section 2.6.1). However in practice it is not really obvious that independence of index terms is a bad assumption.

2.4 Future prospects of the classical models

Comparing the three classical models, we see that the primary factor that differentiates between the Boolean model and the rest is the inability to do the partial matching. Because the Boolean model is not taking into account the partial matching factor it is considered to be weakest of all the three conventional models. But there are variants of the Boolean model which are addressing this inherent weakness in the primitive Boolean model and therefore they are considered worthy. In the next section we will introduce some of the interesting successors of Boolean model.

There has been a lot of work done in the vector and probabilistic models too. New alternatives and new theories are proposed to address the problems in IR. A series of IR models, the successors to the classical models, have been proposed over the years in the research community [Jon; Met04; Tur91; Rob04]. We will look into a couple of the interesting ones in the next few sections and elaborate briefly how the proposed ideas tend to solve the different problems existing in IR and how they inculcate new ideas in IR. The objective of describing different models here is twofold – first to get to know about the different models and how they solve the IR problem, and secondly to acquire the concepts, ideas and theoretical point of views that each of them advocate. Those concepts and theoretical point of views established in different IR models form a strong basis to explore the more complex problems involved in the IR. They also form the basis for the focus of this study, the Link Analysis Ranking models.


2.5 Boolean Based Models

In this section we briefly describe the newer approaches towards the use of Boolean algebra in IR systems.

- Fuzzy Set model
- Extended Boolean IR model

2.5.1 Fuzzy set Model

In the fuzzy set model the document and the query terms are represented through a set of keywords, which are only partially related to the real semantic contents of the respective documents and queries. Therefore in this way we are introducing the notion of partial matching in the Boolean based models. This can be done by defining a fuzzy set for the query terms and the documents are considered relevant to the query if they have a degree of membership in this set. The retrieval process is interpreted in terms of the concepts of the fuzzy set theory. The fuzzy set model is introduced by Ogawa, Morita and Kobayashi [BY99].

Fuzzy set theory deals with the representation of the classes whose boundaries are not very well defined. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition – an element either belongs to or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize the classical sets, since the indicator functions of the classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1. Thus the membership function and core idea of the fuzzy set theory gives a gradual and partial matching capability to the base Boolean formulation.

Despite of the strong theoretic effectiveness, the fuzzy set models for the IR problem has only been discussed in the context of the fuzzy theory in the literature and therefore not a popular model. For more information about the fuzzy set models see [BY99].

2.5.2 Extended Boolean IR Model

There is another alternative model based on the Boolean logic, known as Extended Boolean model for IR problem, introduced by Salton, Fox and Wu [Sal83]. This model is also aimed to relax the strict Boolean logic applied in the primitive Boolean model. It represents a compromise between the strictness of the conventional Boolean system and the lack of structure inherent in the vector-processing system. Hence it preserves the query structure inherent in a Boolean systems and also ranked results of query-document similarity.

The basic idea is that in extended Boolean model the Boolean operations are interpreted in terms of the algebraic distances, normalized Euclidean distance and $L_p$ vector norms for $p \in [1, \infty]$. So, the extended Boolean model tends to exploit the properties of both the set theoretic models and the algebraic models. For more information and details on Extended Boolean IR model, see [Sal83].

Nevertheless, despite its inherent weaknesses, Boolean model has been considered a neat framework, which can be the basis for any other enhanced IR model in present and in future too.

---

2Fuzzy sets are sets whose elements have degrees of membership. Fuzzy sets have been introduced by Lotfi A. Zadeh (1965) as an extension of the classical notion of set.

3An indicator function or a characteristic function is a function defined on a set $X$ that indicates membership of an element in a subset $A$ of $X$. 
2.6 Vector Based Models

In this section we will study a couple of the fascinating vector based models (successors of classical vector model defined in Section 2.3.2). The purpose is to illustrate them briefly and relate them to the focus of study. We will explore the three famous vector based models:

- Generalized Vector Space model
- Latent semantic indexing model
- Neural network model

2.6.1 Generalized Vector Space model

Independence of index term

The assumption of the independence of the index term has been employed in the generalized vector space model. The index terms are expressed in form of vectors, like its predecessor. The assumption of the independence of the index term in vector space implies that the set of vectors (expressing the index terms) are linearly independent and forms a basis for the subspace of interest. The dimension of the subspace of interest is the number of index terms in the overall collection.

Two index term vectors in generalized vector space model might not be orthogonal to each other, which apparently mean that the index term vectors are not seen as the orthogonal vectors which compose the basis for subspace of interest. For that we have: if weights associated with the index terms are all binary then all possible term co-occurrences (inside document) can be represented by the set of power of 2 to the number of index terms in the document, the minterms, given by:

\[ m_1 = (1, 0, \ldots, 0), \ldots, m_i(1, 1, \ldots, 1) \]

where \( t \) is the total number of index terms in a collection.

The main theoretical idea that the generalized vector space model is providing is to introduce a set of pairwise orthogonal vectors \( m_i \) associated with the set of minterms and to adopt this set as the basis for the subspace of interest. And therefore the set of \( m_i \) vectors are then taken as the orthonormal basis of the generalized vector space model.

The independence of index term is now not seen as orthogonality between index terms vectors itself but rather considered to correlate to the minterms \( m_i \) vectors. For example, the vector \( m_7 \) is associated with minterm \( m_7 = (1, 1, 1, \ldots, 0) \) which corresponds to the document in the collection containing the index terms \( k_1, k_2, \) and \( k_3 \). And if such a document is found in the collection than we call the minterm \( m_7 \) as active, and therefore the dependency between the index terms \( k_1, k_2, \) and \( k_3 \) are induced this way.

The idea of the dependency of index term has been independently explored apart from the generalized vector space model (introduced in 1980s). And the basic foundation is the idea that co-occurrence of index terms inside documents in the collection induces dependencies among the index terms. That is why generalized vector space model exploits this basic foundation.

---

4 A basis is a set of vectors that, in a linear combination, can represent every vector in a given vector space, and such that no element of the set can be represented as a linear combination of the others. In other words, a basis is a linearly independent spanning set [wik].

5 For a Boolean function of \( n \) variables \( x_1, \ldots, x_n \), a product term in which each of the \( n \) variables appears once (either complemented, or un-complemented) is called a minterm. Thus, a minterm is a logical expression of \( n \) variables consisting of only the logical conjunction operator and the complement operator. There are \( 2^n \) minterms of \( n \) variables - this is true since a variable in the minterm expression can either be in the form of itself or its complement - two choices per \( n \) variables [wik].
Practical Implications

Despite its nice formulation and strong theoretic relevance it is still empirically unclear that introduction of the term (in)-dependency in IR model would yield any effective improvement. Therefore it unclear that generalized vector model provides any advantage in practical circumstances. In addition, the cost of computing the ranking in the generalized model can be fairly high with large collections because the number of active minterms might be proportional to the number of documents in the collection. For example, in the case of the WWW, the document collection is enormous, billions of documents and it will be highly unpractical and computationally unattainable to do the operations introduced by generalized vector space model.

2.6.2 Latent Semantic Indexing (LSI) model

Inability of index term

Another interesting problem has been addressed in the latent semantic indexing model. The problem concerns the semantic of the index term. Because indexing on the index terms at times give poor results. The fact that the index term is present in the document is considered viable for relevancy, but in some documents that doesn’t always depict relevancy. After all, the index terms are not what the documents are but the ideas and concept present in text represent the document’s identity. It is therefore possible to retrieve a lot of documents that are not quite relevant to the query. And also that there might be quite a few of the relevant documents which are not indexed by the index term, so they might not be retrieved as well. The inherent weaknesses of the index term to represent the content of the document is primarily because of the intrinsic vagueness of IR process based on the keywords. And also because of the inherent problems in the natural language, i.e., the synonymy and polysemy, discussed previously in Section 2.2.

Matching by concepts

Instead of considering index term as a representative element of the document, we need to explore other interesting features of the documents to match with the query. And therefore the ideas in the text more accurately represent which concepts are described in the document. Hence matching under such situations will be, matching the query concept(s) with the documents concept(s), rather than matching with the loosely defined index terms. This will allow documents to be retrieved even they don’t contain the query terms, which is quite a strong capability of an IR model. With such a strong capability the documents can be clustered together according to different concepts and therefore matching will only involve the search across different concepts in the document (concepts) clusters. If a document concept lies under such concept cluster (because it shares the concept), than that document can also be retrieved.

The LSI model is based on the formulation described in the previous paragraph. It tries to solve the issues concerning the index term vagueness. The idea is to map the document and query vectors into a lower dimensional space which subsequently corresponds to different concepts. And indirectly this means that we need to map the index term vectors into lower dimensional space. Therefore the expectation is that the reduction of the dimension produces superior retrieval than the conventional way, i.e., by just index terms.

Formulation

The claims above can be achieved by utilizing the nice results from the Linear Algebra. Consider an association matrix $M$, each entry in that matrix defines the relationship between index terms and the documents in terms
of weights, which can be generated using \textit{tf-idf} (see Section 2.3.2) weighting technique. This association matrix is decomposed using the famous \textit{Singular Value Decomposition (SVD)}\textsuperscript{6} [Lay94] technique (also see Section 3.4.8 for details about SVD). The reduced SVD of the terms \times documents matrix $\mathbf{M}$ is given as:

$$
\mathbf{M}_s = \mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T
$$

(2.5)

where:
- $\mathbf{M}_s$ is the $t \times d$ (terms \times documents) association matrix. $\mathbf{M}_{ij} = 1$ if document $j$ contains $i^{th}$ index term, and 0 otherwise.
- $\mathbf{K}_s$ is $t \times s$ unitary matrix
- $\mathbf{S}_s$ is $t \times d$ diagonal matrix, containing the \textit{singular values} of $\mathbf{M}_s$
- and $\mathbf{D}_s^T$ is $s \times d$ unitary matrix in SVD formulation

Only $s$ largest \textit{singular values} of diagonal matrix $\mathbf{S}$ are kept along with their corresponding columns in matrices $\mathbf{K}$ and $\mathbf{D}$. And therefore the resultant matrix $\mathbf{M}_s$ is the matrix of rank $s$ which is closest to original matrix $\mathbf{M}$ in the least square sense [Lay94]. And $s < r$, is the dimensionality of a reduced concept space. The size of $s$ is controversial; it should be large enough to allow fitting all of the structure of the real data and small enough to filtering out irrelevant representational data.

The relationship between documents in dimension $s$ can be obtained from $\mathbf{M}_s^T \mathbf{M}_s$ matrix. Because matrix multiplication of $\mathbf{M}_s^T (d \times t)$ with $\mathbf{M}_s (t \times d)$ yields $(d \times d)$ \textit{documents \times documents} matrix, hence:

$$
\begin{align*}
\mathbf{M}_s^T \mathbf{M}_s &= (\mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T)^T \mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T \\
&= \mathbf{D}_s \mathbf{S}_s \mathbf{K}_s^T \mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T \\
&= \mathbf{D}_s \mathbf{S}_s \mathbf{S}_s \mathbf{D}_s^T, \quad \mathbf{K}_s^T \mathbf{K}_s = \mathbf{I}_s, \text{ since } \mathbf{K}_s \text{ is a unitary matrix} \\
&= (\mathbf{D}_s \mathbf{S}_s)(\mathbf{D}_s \mathbf{S}_s)^T
\end{align*}
$$

(2.6)

The above matrix quantifies the relationship between each pair of document. Similarly term-term comparisons, i.e., the inner-product of the pairs of term rows, are entries of the matrix $\mathbf{M}_s \mathbf{M}_s^T$:

$$
\begin{align*}
\mathbf{M}_s \mathbf{M}_s^T &= \mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T (\mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T)^T \\
&= \mathbf{K}_s \mathbf{S}_s \mathbf{D}_s^T \mathbf{D}_s \mathbf{S}_s \mathbf{K}_s^T \\
&= \mathbf{K}_s \mathbf{S}_s \mathbf{S}_s \mathbf{K}_s^T, \quad \mathbf{D}_s^T \mathbf{D}_s = \mathbf{I}_s, \text{ since } \mathbf{D}_s \text{ is a unitary matrix} \\
&= (\mathbf{K}_s \mathbf{S}_s)(\mathbf{K}_s \mathbf{S}_s)^T
\end{align*}
$$

(2.7)

All useful comparisons can be made using the rows of the matrices $\mathbf{K}$ and $\mathbf{D}$ appropriately scaled by the diagonal matrix $\mathbf{S}_s$. And these matrices are much smaller than $\mathbf{M}$ (since we assume $s$ as much smaller than $t$ and $d$).

LSI computes the left and right \textit{singular vectors} of the matrix $\mathbf{M}$. Equivalently, the eigenvectors of matrices $\mathbf{M} \mathbf{M}^T$ and $\mathbf{M}^T \mathbf{M}$ respectively. The simplified expressions in equations (2.7) and (2.6) could be used to do that task efficiently.

The main advantage of this model is that it has introduced a nice application of the singular value decomposition into IR problem. The two matrices $\mathbf{M} \mathbf{M}^T$ and $\mathbf{M}^T \mathbf{M}$ are quite useful, we will extensively use them when we study the famous \textit{Link Analysis Ranking} models, such as \textit{HITS}, see Section 3.4. For more details on the LSI model see [Fur88; Coh01].
2.6.3 Neural Network Model

The neural network model for IR utilizes the intrinsic capability of the *artificial neural networks*\(^7\) in order to do efficient *pattern matching*. Pattern matching and pattern recognition is quite central topics in artificial intelligence (AI). In the neural network model of retrieval we employ pattern matching techniques from AI. Since the document vectors and the query vector are extensively compared during the matching process, so it is wise to think of some alternative constructs which can help us form a better relationship and much faster computation of the ranking.

Artificial neural network is composed of *neurons* or nodes which can be viewed as the processing unit (like conventional networks). The output or results from one processing unit (neuron) is fed into another neuron for further processing through the arcs or edges connecting them. The process of sending the results back and forth from neuron to neuron can be repeated until the process *converge* to a good solution. This idea of sending results back and forth is quite similar to the idea of *spread of activation* in network models [AND84].

There is a weight assigned to each *edge* in the neural network. Figure 2.3 graphically illustrates a simple neural network model for IR. The neural network in figure 2.3 contains three layers, query terms, document terms and the documents. The activation or results move back and forth among these three layers. Figure 2.3 seems quite similar to the *inference network* described in Section 2.7.2.

In the neural network, figure 2.3, the *signal or activation* travels from query neurons to document terms.

---

\(^6\)In linear algebra, the SVD is an important factorization or decomposition of a rectangular real or complex matrix.

\(^7\)Artificial neural networks are made up of interconnecting *artificial neurons* (usually simplified neurons) which may share some properties of biological neural networks. Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving traditional artificial intelligence tasks without necessarily attempting to model a real biological system [wiki].
neurons and from document terms to the documents neurons. The query neurons will initiate the signal propagation. However the signal propagation won’t stop after first phase of signal propagation. But the document neuron might as well generate signals which are directed back to the document terms, which is why there are bi-directional edges between documents and document terms neurons. Depending on the signal strength the document term neuron will send back the signal to the document neuron. The process will repeat unless the signal becomes weak enough to halt. This process might retrieve the document which doesn’t even contain the query term, and therefore serve the purpose of being a thesaurus. To make it efficient a minimum threshold level is set such that document nodes below this threshold send no signal out.

The neural network model has not been tested as extensively to see how efficiently it solves the IR problem. But it has introduced another important paradigm in the IR models, and therefore the contribution is quite ingenious. It allows retrieving the documents which doesn’t even match or contain the query term. Therefore it is quite appealing to use such functionalities together with other potential IR models. In the studies [AND84; Car97; Pir96], the idea of dissemination of signal of neural network model has been largely employed to create a better visualization of information on World Wide Web.

2.7 Probabilistic or network based models

Probabilistic models and its variants have been used since at least the early 1960s. Together with the models presented in earlier sections the models in this section form a firm basis for our study. They put forward a lot of interesting concepts and theoretical point of views which are quite necessary to understand the overall IR process in general and Link analysis ranking in particular.

Networks have been used to support diverse retrieval functions, including document clustering, neural network formalism, representation of user knowledge or document content, and also to better match the user interest to the document concepts (e.g., LSI) [Tur91]. The significance of the probabilistic model is more entitled to the formalism of the Bayesian network, which paved the way for the family of the network based models (which is the focus area of this study as well). Bayesian networks are directed acyclic graphical models (DAGs) [Tur91] (see next section).

We will first define the basic prerequisite concepts needed to understand the models. The literature on just probabilistic approaches in IR is by now extensive and at the same time they are highly technical and hard to grasp. In this section we will describe just a few of the relevant ones, in order to capture the new ideas and viewpoints which could help in our study and to better understand and appreciate the problem domain. To comprehend the network topologies in IR and its relevance to the focus area, it is necessary to go a bit more in detail. We first broadly describe the core concepts in network based models and later elucidate on the following model in probabilistic space:

- The inference network model

2.7.1 Basic concepts

Directed acyclic graphs (DAG)

A directed acyclic graph, also called as a dag or DAG, is a directed graph with no directed cycles, i.e., for any vertex \( v \), there is no nonempty directed path that starts and ends on \( v \). DAGs appear in models where it doesn’t make sense for a vertex to have a path to itself, e.g., if an edge \( u \rightarrow v \) indicates that \( v \) is a part of \( u \), than \( u \rightarrow u \) would indicate that \( u \) is a part of itself, which is inconsistent. Informally speaking, a DAG “flows” in a
single direction. In the probabilistic or network based models, the networks are usually considered to be DAG, 
e.g., in case of Link Analysis ranking, self reference (self citation) in the documents (nodes in the network) are 
removed from network, and therefore the webgraph is expected to be a DAG.

Epistemological and Frequentist view

There are two broad categories of probability interpretations which can be called as the physical and the 
evidential probabilities.

The physical probability or the frequentist view takes probability as statistical notion related to law of 
chance. They are associated with random physical systems, e.g., rolling the dice.

The evidential or the epistemological view of probability interprets probability as degree of belief even when 
no random process is involved. It is a way of representing the subjective plausibility of a statement, or the 
degree to which the statement is supported by the available evidence(s). The evidential probability can be 
assigned to any statement, and in daily life we usually use the evidential probability. The probabilistic model 
that we describe here takes the epistemological view of the probabilities.

Bayesian networks

A Bayesian network (or a belief network) is a probabilistic graphical model that represents a set of variables 
and their probabilistic dependencies or independencies [wik]. The Bayesian networks (e.g., figure 2.4) are DAG, 
where the nodes, i.e., vertices represent the random variables, and the edges, i.e., arcs represent the relationship 
between the random variables in the network. The parents of a child node are judged to be the direct cause for 
it. If a proposition represented by the node \( X_1 \) causes or implies the proposition represented by node \( X_2 \), we 
draw directed edge from \( X_1 \) to \( X_2 \) (\( X_1 \rightarrow X_2 \)). The edge between the parent and child nodes characterizes the
2.7 PROBABILISTIC OR NETWORK BASED MODELS

dependency relationship between parent and child nodes.

In general, there are two components which operate independently in such types of graphs: a predictive component in which parent nodes provide support for their children (e.g., the degree to which we believe a proposition, depends on the degree to which we believe propositions that might cause it), and a diagnostic component in which children provide support for their parents (e.g., if our belief in a proposition increases or decreases, so does our belief in its potential causes) [Tur91].

If there is an edge from node $X_1$ to another node $X_2$ than $X_1$ is parent of $X_2$. The set of parent nodes of a node $X_i$ is denoted by $\text{parents}(X_i)$. A DAG is a Bayesian Network relative to a set of variables if the joint distribution of the node values can be written as the product of the local distributions of each node and its parents:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \text{parents}(X_i)) \quad (2.8)$$

The network in figure 2.4 shows the Bayesian network for the joint probability distribution $P(A, B, C, D, E, F)$. Notice that node $A$ has no parent, its local probability is said to be unconditional. To see the dependencies in the network 2.4, and how those dependencies declared in the network are used to calculate the joint probability distribution, see the following formulation:

$$P(A, B, C, D, E, F) = P(A) \cdot P(B|A) \cdot P(C|B) \cdot P(D|B) \cdot P(E|C, D) \cdot P(F|A)$$

The probability $P(A)$ is called the prior probability and can be used to model the previous knowledge about the semantics of the application. Given a set of the prior probabilities for the roots of the DAG, these types of networks can be used to compute the probability or degree of belief associated with the remaining nodes.

The use of Bayesian or inference networks for IR represent an extension to the probability based retrieval models and has been a key success in the IR research [Tur91].

This finishes the brief description of the basic concepts, for more information about the concepts identified here see [Jon; BY99; Tur91]. Now we are in a stage to embark upon the inference network model.

2.7.2 Inference Network Model

Background

The inference network model takes on the evidential probabilistic view of the IR. It is based on the formalism of the Bayesian network. The essence of this approach is to provide a mathematical rule or formalism to explain how to change the existing beliefs in the light of the new evidences. In other words, it allows combining the new data with the existing knowledge.

In general, the inference network model is based on the idea to device fundamental techniques for better understanding the contents of the documents and queries. And that understanding is later used to infer the expected relationship between documents and queries. Based on the inferred relationship, the retrieved documents are presented.

There has been rich amount of research done on retrieval based on inference or evidential reasoning, and there have been a lot of different competing models proposed and developed in this family [Met04; Tur91; ANDS4; Coh01]. The techniques employed in inference networks are in some sense comparable to those used in expert systems, which are devised to reason from uncertain information. The inference network based techniques have therefore roots leading to the artificial intelligence techniques (specifically expert systems).

8In the study of probability, given two random variables $X_1$ and $X_2$, the joint distribution of $X_1$ and $X_2$ is the distribution of the intersection of the events $X_1$ and $X_2$, that is, when both the events $X_1$ and $X_2$ occur together.
The Model

Figure 2.5 illustrates an inference network for IR problem. The model broadly comprises two main components, the document network and the query network. The document network can be built from the document collection using different representation scheme (e.g., from text operations), which might depend on the document corpus and area of interest. The query network depicts the users’ information need and consequently the query representation of that information need. Thus it expresses the user information need based on the system requirement. The document and query networks are joined together by the concept nodes through the links (directed edges) in the network (see figure 2.5).

Notations

A random variable is associated with each node (e.g., $d_j$ for document nodes), represents the event of observing that node. In the case of documents, $d_j$ corresponds to the fact that the document associated with the random variable $d_j$ is being observed in search of the relevant documents. Therefore the observation of a node is the cause for an increased belief in the variable associated with its child node (in case of the document node it’s the index-term or text node $r_m$). All the nodes or the random variables in the inference network are binary variables.

Documents Network

Document nodes correspond to the abstract document representation rather than their physical representation. The number of representations of document in principle are unlimited, in addition to phrase extraction it is expected to have representation based on the natural language processing or automatic keyword extraction. The choice of different types of representation of the documents depends on the requirements of the user and the
specification IR system. The representation schemes can be enforced at the time of the text operation tasks by the document manager, see IR process in figure 2.1.

In addition each document node has a prior probability associated with them, as described in Bayesian networks, which expresses the probability of observing the documents when there are no other evidences available. The prior probabilities are usually assumed to be uniform. For instance, in the original work on inference network the prior probability of observing document $d_j$ is set to $1/N$, where $N$ is the total number of documents in the collection.

$$P(d_j) = \frac{1}{N}$$
$$P(\overline{d_j}) = 1 - P(d_j)$$

Of course there could be other choices for the prior probabilities of the documents, e.g., in tf-idf based ranking strategies, the prior probabilities should be adopted to reflect the knowledge of the importance of document normalization [Tur91; BY99]. In the Link Analysis Ranking models, the prior or initial probabilities are assumed to be in the range of the characteristic polynomial, $\det(A - \lambda I)$ (see Chapter 3). The idea of the prior probability is instructed by the Bayesian postulate, that is, previous knowledge of domain should be asserted in the specification of priors in the network.

Queries Network

On the other hand the query network is inverted DAG, with a single leaf node which corresponds to the event that the user information needs are met (node I in the figure 2.5), and with multiple roots which corresponds to the concepts that express the information need ($q_m$ nodes in the figure 2.5).

In general the user’s information needs are not specifically known and therefore private to the user. The attempt to express the user’s information need in form of multiple query representation is truly experimental and judgemental rather than specific or precise. The representation might depict the natural language formalism (e.g., keywords, phrases), user specific information, e.g., history information, or Boolean like formulation of the queries. At this point we could also instil user personalization factor into the system, which means to cater user specific information to bias the results according to users need (for more details on personalization see Section 4.4). The representation of information need can be undertaken by the text and query operations graphically shown in the IR process in figure 2.1.

The correspondence between the query representation and information need is not precise, but there are query representations which better characterize the user information need, and several query representations taken together for a single information need maybe considered better than individual ones.

The query network (query representations or concept nodes $q_m$) must then be connected to the documents network (document representations or concept nodes $r_m$), depending on some predefined or logical scales of relevance between nodes $q_m$ and $r_m$.

Network behaviour

The documents network is formed once for the given collection and its structure does not usually change during the query processing. In contrast the query network is built for each information need, and it is modified during the query processing as the query is refined and other additional representations are added in an attempt to better characterize the information need. The attachments of the query concepts (mutable) to the document concepts have no immediate effects on the structure of the documents network (immutable).
The documents network is built independent of the query network and therefore is built once. While the query network characterizing the user information need, is built from the available queries, which keeps on refining while the new queries comes in to the system. If the retrieved documents from the available network are inadequate than we need to further elaborate and enrich the query network with more details and specifics in order to better characterize the user information need.

User Information Need

The query nodes $q_n$, represent distinct query forms or representations and corresponds to the events that the query is satisfied. While the single leaf node $I$, in the query network corresponds to the event that the user information need is met. But generally it is not certain that the user information need will be met, it rather depends more on the representations of the documents and the queries.

The overall inference network provides us the facility to combine the information from the multiple document representations and to combine the multiple query representation to form a single, formally justified estimate of the probability that the user’s information need is met [Tur91]. If the characterization of dependencies of the information need on the collection is accurately made, the computed probabilities will provide a good estimation and therefore satisfied users.

The iterative propagation of evidences - from known to unknown

The retrieval process in the inference network based model is intended to capture all the significant probabilistic dependencies among all the random variables represented by the nodes in the network. The inference network calculates the overall probabilistic dependencies iteratively – given the prior probabilities associated with the document (root) nodes and the conditional probabilities associated with the interior nodes, the posterior probabilities or belief associated with each node in the network can be computed or inferred from those probabilities. The “evidences” are propagated from the known nodes (with prior and conditional probabilities) to the unknown nodes (the posterior probabilities) in the network. The network considered this way, represents the dependence of our belief that user’s information need is met on the documents in the collection, while the dependence is mediated by the intermediate documents and query representations.

The initial or seed values associated with each node in the network depict the belief associated with them, e.g., the values associated with the information need node $I$ represent the probability that the information need is met given no specific documents are observed. If any particular document is observed then the values on the different nodes (beliefs) will be recomputed for every node in the network. In particular, the probabilities are recomputed given that a specific document has been observed in the collection. By repeating this process with every document in the collection we can compute the probability that information need is met given that all the documents in the collection have been observed, and therefore rank the retrieved documents based on the observations data.

Formulation

There could be different techniques for finding the relevance of the documents to the information need. For example, the simplistic way is that each document is considered in isolation. But considering a subset of the documents might produce higher probability that information need is met. The techniques like clustering, classifying, grouping together related documents and any other sophisticated approach could be used for relevancy. But here we stick to the simplistic way of calculating the relevancy.
The ranking of a document \( d_j \) with respect to the query \( q \) is a measure of how much evidential support the observation of \( d_j \) provides to the query \( q \). In the inference network, the ranking of a document \( d_j \) is computed as \( P(q \land d_j) \) where \( q \land d_j \) are equal to 1. Let \( k = (k_1, k_2, \ldots, k_t) \) is a binary random variable, i.e., \( k \in \{0, 1\} \), we have:

\[
P(q \land d_j) = \sum_{\forall \vec{k}} P(q \land d_j \mid \vec{k}) \times P(\vec{k})
\]

\[
= \sum_{\forall \vec{k}} P(q \land d_j \land \vec{k})
\]

\[
= \sum_{\forall \vec{k}} P(q \mid \vec{k}) \times P(d_j \mid \vec{k}) \times P(\vec{k})
\]

\[
= \sum_{\forall \vec{k}} P(q \mid \vec{k}) \times P(\vec{k} \mid d_j) \times P(d_j)
\]

\[
P(\overline{q} \land \overline{d_j}) = 1 - P(q \land d_j)
\]

which is obtained by application of the Bayes’ rule [Lay94]. \( \overline{q} \land \overline{d_j} \) denote \( \neg(q \land d_j) \). Going into further details of relevancy calculation is out of scope of this study, for further details see [Tur91; Met04].

**Link Matrix**

Nodes often connected to multiple parents have to contain functions for computing the probability of their child nodes based on the probabilities of all the parent nodes (which are known). A direct way of encoding the estimated probabilities of all non-root nodes can be done by link matrix. Link matrix is also used extensively in Link Analysis Ranking models (see Chapter 3).

This matrix can be of size \( 2 \times 2^n \) where \( n \) is the number of the parents. They define node probabilities in dependency of parent states that are certain, i.e., parent node probabilities of 0 or 1 not in between. But this encoding is dependent on the number of parents. Encoding on the link matrix is practical only, when the number of the set of the parent nodes is small. So, the estimation of the probabilities has two main parts: how to estimate the dependence of a node on its set of parents and how to encode the estimates in a usable and efficient form.

In literature mostly the link matrix form is used to encode the estimates. There are different canonical link matrix forms for the inference networks to better capture the two main parts described above for the estimation of the probabilities in the network. For more information about the link matrix forms, see [Tur91].

The usage of different link matrices implementing the network’s nodes allows implementing features of Boolean based models, vector based model, and probabilistic models. Therefore, inference networks can for example rank documents using Boolean query syntax [Bau01]. InQuery is a functional IR system based on the inference network model [Met04].

**Computational complexity**

The cost of computing the ranks is linear on the number of documents in the collection. Therefore the cost of computing an inference network ranking has the same complexity as the cost of computing a vectorial ranking [BY99]. The Bayesian network model described above does not impose any significant additional cost for computing ranks. This is due to the structure of the network; it is DAG, which means there aren’t any cycles in the network, so there is no unnecessary looping. This implies that the belief propagation can be done in a time proportional to the number of nodes in the network.
The Bayesian network based models provide a framework which allows neat combination of distinct evidential sources to support a relevance judgement on a given document [BY99].

2.8 Structured and Un-structured document Retrieval Models

2.8.1 Background

The internal structures (sections, subsections, chapters, etc) of the document are very important properties of the documents. Apart from their inherent peculiarities, it is also necessary to take them into consideration when the document corpuses contain structured documents. IR model should therefore exploit the internal structures of the documents in order to offer improved and contextual relevancy ranking. There has been some research done on figuring out whether the document’s internal structures do constitute any important source of information for relevancy [Rob04].

It is therefore quite beneficial and sometimes critical to exploit the documents internal structures to enhance relevancy and hence enrich the usability in IR. The IR system built entirely for structured documents is objectively different then IR system built for un-structured documents. At times users are eager to search through the structures of the documents. A query such as, search for word ‘information retrieval’ in the abstract, subsection, or section of a chapter is highly expected. And even more, a user might require having a visual location of the query string in the document. For example, for the query ‘information retrieval’, a snapshot of the location where a match has found could suffice. Such requirements cannot be effectively fulfilled by the conventional retrieval models, unless some measures are taken. The visually and structurally manipulated search results can only be achieved in IR systems which are built on the idea of structured document retrieval.

The structured document retrieval might illustrate the appeal of a query language which allows us to combine the specification of strings (identifying the visual and structured information) with the specification of structural component of document. Structured document retrieval can therefore be thought of an IR model which tends to combine the information on contents with information on document's internal structures.

2.8.2 Brief Taxonomy

There has been a series of IR models proposed over the years which tend to focus on structural aspect of documents during retrieval. Various structured retrieval models have appeared throughout 1990s in the literature [BY99]. We will briefly define a few of the relevant ones here, and focus on characterizing and conceptualizing their strong points.

The most ad-hoc retrieval systems apply standard (non-structured) ranking algorithms and tackle the structured documents by combining in some way the scores obtained from different fields (abstract, chapter, sections, subsections etc). One of the earliest empirical research in the area of field weighting is the work of Wilkinson [Rob04]. He evaluates different ways to weight and combine the scores obtained on different fields of a document. In practice, many systems exploits structure in an ad-hoc manner, by implementing a linear combination of the scores obtained from scoring every field. There have been discussions on how to combine scores efficiently, how are scoring done in terms of simplicity and interpretation, and the computational complexity in terms of accuracy and performance issues involved in different scoring mechanisms. The different methods defines a mapping from structured to non-structured documents, the method can be applied to any ranking function for non-structured documents [Rob04].
2.8.3 BM25

BM25 is a method for ranking structured documents. It is based on 2-Poisson model of term frequencies in documents [Rob04], which could be seen as elementary form of unigram language model, with the model parameter for a given term in any document depending on a single binary hidden variable known as ‘eliteness’. For each term, the collection of document is split into two classes, elite and non-elite. It could be seen as, for any given document the terms are classified into elite and non-elite. Elite terms are those which appear in the more important part of the document, for example, in the title or abstract. Thus each term occurrence in such a field can be taken as stronger evidence of the eliteness of that term in the document than an occurrence in the body. For more information about BM25 scheme see [Rob04].

2.8.4 Anchor Text

Anchor text is a text associated with the link in a source document, which is assumed to describe the target document. Anchor texts are widely used in the documents available on the Web. Apart from its inherent capabilities, the presence of the anchor text in the structured documents also triggers some problems in retrieval. The target that anchor text points to is to be extracted from its source and embed in to the target. One of the problems is that, the source pointing to by the anchor text is not usually written by the same author, and now forms part of the target text. In addition there will be repeated and nested fields in the document (multiple abstract, repeating chapters, etc).

In the study by Chakrabarti et al., [Cha98], an automatic resource compilation uses text in the vicinity of href (i.e., the anchor text) in the document as a descriptive of the contents present in the target document. Using the weights of terms in the anchor text (the vicinity of href) together with the hyperlink information (e.g., by using HITS idea, see Section 3.4) to compute and return a list of web sources are considered to be the most relevant to the query topic.

Discussion about the ways of dealing with the anchor text problems is out of scope of this study, for more information about the anchor text and problems that it brings in see [Rob04; Cha98].

2.8.5 Only Structured Documents

Thinking of the document corpus to have (or consider) entirely structured documents would lead us to approaches that mainly manipulate the structure of the documents. Burkowski [BY99], proposes a model based on non-overlapping list. Divide the whole text in each document in non-overlapping text region which are than collected in a list. Since there will be multiple ways to divide a text in non-overlapping regions, so, multiple lists are generated. The text regions in the same list have no overlapping, the text regions from the distinct lists might overlap.

Based on the above model the indexing subsystem must also be accommodated with the structural information and therefore structural component stand as an entry in the indices [BY99].

There could be some efficiencies done to the approaches defined above, for example each of the non-overlapping lists composed of chapters, sections, paragraphs, etc, we could exploit the notion of nearest neighbour or proximity to do an efficient retrieval and more expressive queries. More complex models for structured retrieval have also been proposed in the literature [BY99].

---

9href is used in a HTML tag. It indicates the URL being linked to and makes the anchor into a link. For example, this tag creates a link to homepage.html: HREF="www.homepage.html"
2.9 Summary and reflections

Traditionally three main entities can benefit from the research in IR models: library systems, specialized retrieval systems (e.g., ACM, IEEE, etc) and foremost the Web (WWW). But as there is a much higher focus on the retrieval process largely, there is much higher tendency in most sectors where IR models can carry positive contributions. For instance, the growing need for the media and the growing number of varieties led to increasing need of a retrieval system which can facilitate to choose. There has been research going on in the area of information filtering which provides recommendations, using content-based and collaborative filtering techniques [Bau01].

Library systems being one of the motivators for the need of IR systems, still provide fuel to the research in IR models and therefore the advent of the digital libraries (with more diverse repositories) have more direct dependencies on efficient IR techniques.

The case of World Wide Web is quite different and unique. The WWW became the ultimate signal for the dominance of the information age. Dealing with the highly dynamic source of information and at the top of it the exponential growth of information has really become a challenge. And having such a huge and dynamic corpus of documents and doing operations like querying and searching on such huge amount of data apparently floundered the user in general and research communities in particular. But there have been remarkable amount of work done throughout the history of IR, for example in 1998 all this thoughts changed, when link analysis hit the IR scene [Bri98; Kle99]. Web search improved dramatically, and web searcher rigorously used the search engines. The addictive and obsessive use of the search engines made them principally a crucial part of the web. The focus of our study is also the Link Analysis based IR models (see next chapter). Application of the core concepts of Linear Algebra in Link Analysis Ranking (LAR), especially the most celebrated application of Singular Value Decomposition, as is used in Latent Semantic indexing model, and theory of Markov chains have become the major reasons for the success of LAR.
Chapter 3

Link Analysis Ranking

3.1 Link Analysis Ranking (LAR)

The exponential increase in the size of Web is now very eminent. Just the amounts of only textual information are estimated to be in order of hundreds of terabytes. Web is seen as very large, diverse, unstructured but ubiquitous databases of information. Subsequently it triggers the need for efficient tools to manage, retrieve and filter information from these databases.

The apparent ease with which the users click from document to documents provides a rich source of information which could be used to understand what and where to find the important documents. The unstructured and diverse collections of documents are held together by the billions of annotated connections called hyperlinks. Analyzing these myriad interconnections between the documents forms the basis for Link Analysis Ranking.

Research in Link analysis ranking is objectively derived from Bibliometrics research, the analysis of the citation structure among academic papers. In Bibliometrics research, the citation structure of a body of document is used to produce numerical measures of the importance and impact of papers [Lar96; Lan06].

Simple and explicit heuristics and methods might not be able to take advantage of the implicit structure of documents where they reference each other, and therefore, provide a wealth of information. The simple and conventional heuristics rank the pages by the occurrence of query term or any other content based methods. These heuristics are intrinsically susceptible to be misused, e.g., by spamming. Spamming is the practice of eliciting favourable rankings, by designing the documents in such a way as to boost their ranking score. One of the main reasons of extensive research in the area of Link Analysis Ranking is to prevent spamming [Lan06].

Not only that the conventional text based methods face problems inherent in the human language, i.e., the problem of Synonymy and Polysemy (as described in Chapter 2). They also need to cater the changing needs of the users by providing them much relevant results from the huge and dynamic sources. With these problems intact an IR system would end up having frustrated users, not satisfied with the search outcomes.

The citation structures of the documents contain a wealth of useful but implicit information. Through citation structure hundreds and millions of documents can be pulled together into a network of knowledge. Foremost such a structure represents the users’ behaviour and need. The users discover most relevant and valuable information through recommendations and references from a good source of information.

LAR research mostly tends to form a visualization of information, explicitly or implicitly, for making sense of information sets with thousands and millions of objects [Car97]. Through visualization or maybe clustering we can have an overview of the locality and connectivity of information on the Web.
It is the concept of *recommendations* and *endorsements* that exist within the documents in a form of hyperlinks or citations that have been mainly exploited and focused upon in LAR. The *spreading of endorsements* from node to node in the network of documents plays vital role in deciding the importance of documents to the query topic. Different LAR models treat the spread of endorsement in numerous ways. The spread of endorsements in LAR is influenced from the use of *spread of activation*\(^1\) in neural networks–artificial intelligence (see Sections subsec:neuralnetwork).

There are studies in LAR which are primarily based only on the connectivity analysis, and there are also other studies that analyze both the content and connectivity information for utilizing the idea of spread of endorsements [Kai98; Car97]. In the study by Piroli et al., [Pir96], together with content and connectivity analyses, they also consider a graph structure representing the flow of users from node to node, the edges representing the number of users that go along them.

The motivation and wealth of research in LAR is due to the massive size of webgraph and the huge amount of *latent information* available in that graph which could be used in various occasions. Influenced primarily by the usefulness of webgraph and subsequently by other related researches such as [AND84; Pir96; Car97], the seminal work of Brin and Page 1998 [Bri98] and Kleinberg 1998 [Kle99] became the pioneers who formally introduced LAR. The ingenuity was mainly because of the large scale application of core concepts in *linear algebra*.

*Hyperlink structures* are then used to determine the relative *authority* of the documents and produce improved algorithms for the ranking of search results [Bor05]. Most of the research in LAR is primarily carried out to elaborate the mathematical concepts or improve the ideas initially proposed by Kleinberg, Brin and Page. We will discuss in detail the algorithms developed by them in this chapter and also explore other relevant LAR models. There are also other group of researchers focusing on the LAR problem independent of the two seminal works, who intend to explore any other formulation of LAR, such as analyses of the pure graph structure of web (e.g., *Power-law distribution* formulation of the webgraph) [Bro00; Cha02; Lu04].

In the Chapter 2 we have explored various models in IR in a general setting, but in this chapter the intention is to explore IR models in the more focused settings. Hence we consider documents interconnected with each through hyperlinks or citations. That is, the focus is to independently explore Link Analysis Ranking models. We start with a general model and than illustrate the other focused approaches.

### 3.2 InDegree Algorithm

One of the general formulations could be that, a *good* document is the one which is pointed to by or cited by many other documents in the network of documents. The number of *in-links* to a page provides a measure of its popularity and quality. The *InDegree* algorithm can be considered as one of the *primitive* algorithms in LAR, that could be used to rank the pages according to their *popularity* or *in-degree*. As the name of the algorithm suggests, it measures the number of documents that link to a document, therefore, it ranks the documents according to their in-degree in the graph. Hence for every node \(i\), we have:

\[
 a_i = |B(i)| \tag{3.1}
\]

This simple heuristic is applied to several IR systems in the early days of the web search [Mar97].

---

\(^1\)“Spread activation is the process by which activation spreads from node to node along network links, making knowledge associated with particular sources of activation (i.e., foci of attention) available for processing” [AND84]. Through spread of activation process one can identify knowledge relevant to current focus of attention.
3.2.1 Relevance to other Models

Kleinberg [Kle99] doesn’t seem to be convinced by just the InDegree heuristic and therefore considers InDegree algorithm not sophisticated enough to capture the importance or authoritativeness of the documents on Web. And Page et al., [Bri98] extends the idea of InDegree by observing that all the links does not carry the same weight, and therefore proposed a new definition of importance, by saying that a “good” document is the one which is pointed to by many other “good” documents. And when Lempel et al., [Lem00] sashayed into the game, they blend together the ideas presented earlier by Kleinberg, Page and Brin.

If the web graph $G$ has just one connected component, i.e., the underlying Markov chain (see Section 3.3.4) is irreducible, then the algorithm proposed by Lempel et al., the SALSA, reduces to InDegree algorithm [Bor05]. Furthermore, even when the graph $G$ is not connected, if the starting point of the random walk (see Section 3.3.4) is selected with probability proportional to the “popularity” (in-degree) of the node in the graph $G$, then their algorithm again reduces to the InDegree algorithm. This algorithm was referred to as PSALSA (popularity-SALSA) by Borodin et al., 2001 [Bor05].

The purpose of defining InDegree algorithm is also to see the resemblance of the modern LAR models with this primitive model. Hence in experiments we found that there are some of the new approaches in LAR whose top − 15 results sufficiently look like the top − 15 outcomes of this primitive algorithm. Though they differ in the way they calculate the rankings but the outcomes are observed to be similar in the prescribed settings (see Chapter 5).

3.3 PageRank

PageRank algorithm for calculating the relevancy of the pages has been a landmark in the field of Link Analysis Ranking. The seminal paper of Brin and Page [Bri98] introduced the initial idea of PageRank, which exploits the hyperlink structure of the Web to determine the relative popularity or importance of each webpage. The hyperlink structure of Web is used to build a Markov chain formalism of IR, by iteratively calculating the stationary distribution vector until the rank vector is found, independent of any queries. The approach is novel primarily because of the relevant and prospective application of Markov chain theory.

“The heart of our [Google.com] software is PageRankTM …” [goo], PageRank is thus considered as an integral part of the Google’s success.

In this section we will explore the PageRank algorithm in terms of core concepts used in the initial scheme and its inherent limitations. We will study the algorithm with the help of suitable examples with necessary details. The purpose is to study different properties and peculiarities of the algorithm and its implication on overall LAR research subsequently its importance and eminence to the focus of our study.

3.3.1 Web Graph

Given the document corpus and index structure containing hyperlink information, the first step is to construct the underlying graph from the hyperlink structure in documents. The hyperlink structure of the documents on the Web, form a massive directed graph. The node in such a graph represents the webpages (documents), and the edges represent the hyperlinks. The graph is directed because the edge from one node $P_A$ to another node $P_B$ ($P_A \rightarrow P_B$) means that the page $P_A$ contains the hyperlink to page $P_B$. The webgraph is simple in a sense that if there are multiple links from one page to another, only a single edge is placed between them in appropriate direction.
Links within the same webpage are removed (e.g., navigational links, such as, ‘back’, ‘home’). The links into a webpage are called in-links while the links out of a webpage are called out-links. Every node has a degree, similar to the graph theoretic formulation. Degree refers to the number of edges incident to a node (in-links and out-links). Hence the total number of links going out of a node are called out-degree, while the total number of links coming into a node are called in-degree of a node in the webgraph.

![Figure 3.1: The webgraph or the internet map - Image taken from [com]](image)

Extracting meaningful information from the massive webgraph is not that straightforward, because the nodes and edges create a very chaotic and dense graph. Drawing the webgraph alone is a Herculestic task, if drawn it will be the largest human artefact ever created. Webgraph the way it is impossible to visualize unless some clustering algorithm is used to make it visually friendly. See for example figure 3.1 for the clustered view of the webgraph. It is quite problematic to disentangle such a complex graph in order to even present it in a meaningful way. But there has been some work done such as, the Atlast of Cyberspace [Dod01] presents over 300 colourful informative maps of cyberactivities. For more information on structure of web see [Hir00; Bro00; Ara01].

There is also an extensive amount of graph theory involved in the Link Analysis Ranking. Because we exploit the massive webgraph, and to extract meaningful and relevant information from such a massive graph requires a much closer understanding of the underlying graph structure and realization of the network topology involved.

The problem of understanding the input graph is of fundamental importance for the study of LAR models. Undoubtedly we need to understand how the structural properties of the graph affect the ranking of the algorithm [Bor05].

One of the interpretation of the hyperlinks in webgraph is to consider a hyperlink from page $P_A$ to page $P_B$ as an endorsement given from page $P_A$ to $P_B$. Academic citations in literature or “bibliometrics” of academic
papers broadly support the idea of endorsements. The “bibliometrics” of the Web hence is largely employed to account for the citations (endorsement) bestowed to each webpage by accumulating their in-links. That is, hyperlink in the documents are considered as a recommendation, an edge from pages $P_A$ to $P_B$ means $P_A$ recommends page $P_B$, and therefore $P_A$ consider page $P_B$ as an important page, quite similar to academic citations.

### 3.3.2 Google’s PageRank

The Google’s PageRank considers the endorsement (edges) slightly different, the status of recommender is also considered important here. A recommendation or hyperlink from one page to another page is considered good if the recommender itself is a good source, which means that the recommender has a high PageRank itself. In a nutshell a webpage is important if and only if it is pointed to by other important pages [Bri98].

In some texts, the recommendation is identified as vote, and therefore Web becomes a democracy where pages vote for the importance of each other by hyperlink towards one another [Bry06].

The idea described above seems self-referential. There is something more than just vote or recommendation. A single page might illicitly gain influence by recommending a lot of other pages, but we don’t want this to happen. Therefore curbing such an effect by evenly dividing the importance or PageRank of the recommender among all of its outgoing links.

In essence the amount of importance conferred on $P_k$ by $P_j$ ($P_j \rightarrow P_k$) is directly proportional to the importance of $P_j$ and inversely proportional to the number of pages $P_j$ points to. Thus, if page $P_j$ contains $|P_j|$ links, one of which links to page $P_k$, then page $P_k$’s score will be boosted by:

$$\pi(P_k) = \sum_{P_j \in B_{P_k}} \frac{\pi(P_j)}{|P_j|}$$  \hspace{1cm} (3.2)

where $\pi(P_k)$ is PageRank score of page $P_k$ and $B_{P_k}$ is the set of back-links (in-links) of page $P_k$.

Now the PageRank of $P_j$ is evenly distributed among the pages linked by it (i.e., the set out-links of $P_j$). But to compute the equation (3.2) we need to know PageRank of $P_j$, i.e., the PageRank of in-linking pages to page $P_k$. The PageRank of the in-linking pages can be calculated in an iterative fashion. Starting with an initial PageRank values for each page (the seed values) in corpus, e.g., $1/N$, where $N$ is total number of pages in the corpus, the iterative algorithm will compute the new values of the PageRank successively from the previous values. Thus the equation above can be written in iterative form as:

$$\pi^{i+1}(P_k) = \sum_{P_j \in B_{P_k}} \frac{\pi^i(P_j)}{|P_j|}$$  \hspace{1cm} (3.3)

where $\pi^{i+1}(P_k)$ is the value PageRank of $P_k$ at time step $i + 1$.

The process starts with some initial value, and successively compute the new values assuming that with such an approach the PageRank scores will eventually converge to some stable and desirable final values.
3.3.3 Link Matrix

Let us consider an example to illustrate the concepts and ideas described above. The figure 3.2 can be represented in a link matrix form as follows:

\[
A = \begin{pmatrix}
0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0
\end{pmatrix}
\]

The link matrix \( A \) above is a row normalized matrix with, \( A_{ij} = 1/|P_i| \) (\(|P_i|\) is the out-degree of page \( P_i \)) if there is a link from node \( i \) to node \( j \), and 0 otherwise. The matrix \( A \) is similar to the adjacency matrix in graph theory; except its nonzero elements represent the endorsements (e.g., through probability). At any iteration \( t \), the next successive values of PageRank \( t+1 \) for above graph can be written in a system of linear equations as:

\[
p_{1}^{t+1} = 1/2p_{1}^{t} + 1/3p_{4}^{t} + 1/2p_{5}^{t} \\
p_{2}^{t+1} = \frac{1}{2}p_{2}^{t} + p_{3}^{t} + \frac{1}{2}p_{4}^{t} \\
p_{3}^{t+1} = \frac{1}{3}p_{1}^{t} + p_{3}^{t} + \frac{1}{3}p_{4}^{t} \\
p_{4}^{t+1} = \frac{1}{2}p_{1}^{t} + \frac{1}{3}p_{4}^{t} \\
p_{5}^{t+1} = \frac{1}{2}p_{5}^{t}
\]

The above linear system of equations can be written in matrix form as:

\[
x = xA \text{ where } x = [p_1, p_2, p_3, p_4, p_5]
\]

(3.4)

The equation (3.4) seems quite familiar in term of linear algebra formulation. The problem of finding the PageRank is now transformed to “standard” problem of finding the eigenvectors of the link matrix \( A \).

Thus equation (3.4) will iteratively compute the PageRank \( n \)– vector \( x \) of graph (\( n = 5 \) in figure 3.2). Instead of solving the system of linear equations we can solve the matrix formulation. In terms of matrix notation, the PageRank vector can be represented as a row vector \( \pi \) with \( k^{th} \) iteration as a superscript as follows:

\[
\pi^{(k+1)} = \pi^{(k)}A
\]

(3.5)
Considering the structure of the link matrix $A$, the $i^{th}$ row of $A$ contains $|P_i|$ (the number of out-links of page $P_i$) nonzero entries, each equals to $1/|P_i|$, and therefore the row sums to 1. This motivates the observation in the next section.

### 3.3.4 Markov chain and Random Surfer (Walk)

![Figure 3.3: Network of five documents containing “dangling nodes”](image)

Brin and Page use the notion of a random surfer (synonym of random walk in Markov chain theory [Bré99]), to describe the process of calculating the PageRank of the whole web. The random surfer is given a webpage at random and he keeps following the hyperlinks, never clicks ‘back’. The surfer hops along randomly chosen hyperlink structure of the web. When he arrives a webpage with a lot of outlinks he chooses one of them randomly, and continues with these random decisions indefinitely unless theoretically all the pages are exhausted (visited).

The relative importance of the webpages (PageRank) can be viewed as the proportion of time the random surfer spends on a webpage. If he spends large proportion of his time on that page, he must have repeatedly found himself returning to that page, when randomly following the hyperlinks. But the random surfer eventually gets bored and randomly starts on another random page (e.g., by writing in the browser). PageRank, therefore can be thought of a model depicting user behaviour [Bré98].

It is interesting to note that the behaviour of the random surfer can be depicted through the application of “Markov chain” (i.e., the sequence of random walk). A random surfer will have a random walk on the Web in a form of navigation between pages, each page represents a possible state, and each link represents a possible transition. And therefore the random walks by the surfer is Markovian because transition of each step is independent of the previous steps and it only depends on the current state [Bré99] (also see [Mot95, 132]). Hence the PageRank vector $\pi$ represents the stationary distribution or equilibrium distribution of a random walk on the entire Web.

In the study of Markov chains, a square matrix is called row-stochastic matrix if all of its entries are nonnegative and the entries in each row sum to 1. The matrix $A$ of figure 3.2 is a row-stochastic matrix, while
the matrix corresponding to figure 3.3 is almost a row-stochastic matrix but it contains **dangling nodes**.

\[
A = \begin{pmatrix}
0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0
\end{pmatrix}
\]

The nodes with no outlinks, which create 0 rows in the matrix are called **dangling nodes**. But all other rows corresponding to non-dangling nodes, therefore creates row-stochastic, thus \( A \) of figure 3.3 is a **substochastic** matrix.

### 3.3.5 Limitations and Adjustment

The presence of the dangling nodes in the webgraph create some problems for the random surfer, whenever he enters a dangling node he gets stuck (there are no out-going links). Considering the fact that there will be plenty of dangling nodes on the web, e.g., pages that do not cite other pages, the random surfer must do something to get rid of them. According to theory of Markov chain, the link matrix must be row-stochastic, i.e., the row sum of the link matrix \( A \) must be equal to 1. But with existence of dangling node, the row sum of \( A \) will be 1 but will also be 0 in case of dangling nodes.

In order to fix this problem we need to have an adjustment, the \( 0^T \) rows of the link matrix \( A \) is replaced by \( \frac{1}{n}e^T \). Thus making matrix \( A \) as row-stochastic. By this rank-one update of the link matrix \( A \) we will get a new matrix:

\[
S = A + a\left(\frac{1}{n}e^T\right)
\]

where \( a \) is a binary vector called **dangling node vector** with \( a_i = 1 \) if page \( i \) is a dangling node (with no out-links) and 0 otherwise.

But still there is another problem; to uniquely compute the PageRank vector the webgraph must be **strongly connected**\(^2\). Unfortunately it is not always possible that the link matrix \( A \) will yield a unique PageRank of the webpages. If the enormous webgraph is not connected, containing many **disjoint components** (which is usually the case), it will create difficulty to compute the unique ranking of the pages. According to **Ergodic theorem** for Markov chain, the stochastic matrix \( A \), should also be **primitive** in order to converge to a **unique stationary distribution vector** (PageRank vector \( \pi \)) [Mot95]. If all states in a Markov chain are Ergodic the PageRank vector will be unique. A primitive matrix is both **irreducible** and **aperiodic**. A matrix is irreducible if the graph that this matrix represents is connected, every node is connected to every other node, while aperiodic matrix is the one with **self-loops**.

In order to do another adjustment to the modified link matrix \( S \), we need to observe the behaviour of the random surfer (the web user). The random surfer at times gets bored, and abandon the hyperlinks surfing, and therefore randomly jumps to another URL e.g., by entering new destination in the browser. This behaviour of the random surfer is recognized as “Teleportation” [Bri98]. When a random surfer teleports, he begins again to surf the hyperlinks, and again gets bored and do another teleportation and so on. This behaviour of the random surfer can be mathematically represented with following matrix:

\[
G = \alpha S + (1 - \alpha)\frac{1}{n}ee^T
\]

\(^2\)In graph theory, a directed graph is called **strongly connected** if for every pair of nodes \( P_A \) and \( P_B \) there is a directed path from \( P_A \) to \( P_B \) and \( P_B \) to \( P_A \).
where $0 \leq \alpha \leq 1$, it’s a parameter that controls the proportion of time the random surfer follows the hyperlink as opposed to teleporting. If $\alpha = 0.85$, than 85% of time the random surfer follows the hyperlink structure of the Web, and 15% of time he teleports.

The stochastic matrix $G$ is the weighted average or more closely the convex combination of the link matrix $S$ and $E = 1/n e^T$. Notice that matrix $G$ is now primitive which guarantees a unique and positive PageRank vector $\pi$ for the entire webgraph. If $G$ is primitive then Ergodic theorem guarantees that any iterative method applied to the equation (3.6) will converge to a unique and positive PageRank vector (the stationary distribution of the random walk) [Mot95; Hav03b].

But due to the above adjustments now the modified link matrix $G$ is completely dense, as opposed to the original link matrix $A$ which was quite sparse. Fortunately we could write $G$ in terms of the original matrix $A$, which will yield some computational gains.

\[ G = \alpha S + (1 - \alpha) \frac{1}{n} e e^T \]  \hspace{1cm} (3.7)

\[ = \alpha (A + \frac{1}{n} a e^T) + (1 - \alpha) \frac{1}{n} e e^T \]  \hspace{1cm} (3.8)

\[ = \alpha A + (\alpha a + (1 - \alpha) e) \frac{1}{n} e e^T \]  \hspace{1cm} (3.9)

The above matrix $G$ (after two adjustments of matrix $A$) turns out to resolve the problems identified, and therefore provides a unique PageRank vector $\pi$ for the entire webgraph. Thus we have

\[ \pi^{(i+1)} = \pi^{(i)} G \]  \hspace{1cm} (3.10)

### 3.3.6 Power Method

The iterative nature of the PageRank computation leads to the choice of different methods to solve it. The available iterative methods could be the famous Gauss-Seidel, Jacobi or GMRES, BICGSTAB, etc [Bar94]. We have the following linear system:

\[ \pi^{i+1} = \pi^i G \]

\[ \pi^T e = 1 \]

The goal is to determine the dominant (principal) left eigenvector of $G$ corresponding to the dominant eigenvalue $\lambda = 1$. The normalization equation $\pi^T e = 1$ ensures that $\pi$ is a probability vector as $e$ is all ones vector. $\pi$ is stationary vector of Markov chain with transition matrix $G$, there has been wealth of research on the numerical solution of the Markov chain problem [Bré99; Lan06], they contain over a dozen methods to find the stationary vector $\pi$. But the oldest numerical method for computing the principal eigenvector of transition matrix, the Power method, conforms more to the peculiar nature of the webgraph’s stochastic matrix $G$. Power method starts with an initial vector $\pi^0$, then generate the sequence $\pi^k = \pi^{(k-1)} G$, ($\pi^k = \pi^0 G^k$) where $k$ approaches $\infty$ ($\lim_{k \to \infty} \pi^{(k)}$), where $\pi^{(k)}$ is the stationary distribution of Markov matrix $G$.

In linear algebra, the power method is an eigenvalue algorithm, given a matrix $A$, the algorithm will produce a number $\lambda$ (the eigenvalue) and a nonzero column vector $\vec{v}$ (the eigenvector), such that $A \vec{v} = \lambda \vec{v}$.

The intuition behind convergence of Power method is as follows. Assume that the initial vector $\pi^{(0)}$ lies in the subspace spanned by the eigenvector of $G$. Now $\pi^{(0)}$ can be written as a linear combination of the eigenvectors of $G$:

\[ \pi^{(0)} = \alpha_1 \vec{u}_1 + \alpha_2 \vec{u}_1 + \ldots + \alpha_m \vec{u}_m \]  \hspace{1cm} (3.11)

And because the principal or first eigenvalue of the transition matrix of Markov chain $G$ is $\lambda_1 = 1$

\[ \vec{x}^{(0)} = G \vec{x}^{(0)} = \vec{u}_1 + \alpha_2 \lambda_2 \vec{u}_1 + \ldots + \alpha_m \lambda_m \vec{u}_m \]  \hspace{1cm} (3.12)
Hence,
\[ \tilde{\pi}^{(k)}T = G^k \tilde{\pi}^{(0)}T = u_1 + \alpha_2 \lambda_2 u_1 + \ldots + \alpha_m \lambda_m u_m \] (3.13)

Since \( \lambda_n \leq \ldots \leq \lambda_2 < 1 \), \( G^k \tilde{\pi}^{(0)}T \) approaches \( u_1 \) as \( k \to \infty \). Thus power method ensures convergence to the principal eigenvector \( u_1 \) of the transition matrix \( G \).

Compared to other iterative methods, power method is generally the slowest method. Considering the equation (3.9), where \( G \) is expressed in terms of the sparse matrix \( A \), the power method turns out to be quite appropriate [Lan06]. Power method is at the heart of both the motivation and implementation of PageRank algorithm [Hav03a]. Essentially power method is a matrix-free, which means there are less concerns about the handling of the storage issues because of the huge matrix \( G \). There is no extensive manipulation of matrix done in the power method. And we do not store much in each iteration; just the sparse matrix \( A \) and the dangling node vector \( a \) and only one more vector \( \pi \). Other iterative methods requires storage of at least 10 vectors each iteration [Lan06; Lay94], which in case of the enormous size of the Webgraph is quite expensive. The rate of convergence of the power method is better than other methods, it’s hard to find a method that can beat 50 \( O(n^3) \) of power iterations [Lan06].

### 3.3.7 Effects of Random Jump

The idea of random jump at the time of teleportation by random surfer provides additional flexibilities to PageRank. Depending on the teleportation probability \((1 - \alpha)\) the surfer visiting any node will jump to a random page. Random jumps offer multi-faceted benefits for overall success of PageRank. The random jumps were primarily employed to guarantee the irreducibility of the web’s transition matrix. But this model could be used to adapt the PageRank algorithm in numerous situations.

By controlling the random jumps of the surfer we could bias the ranking produced by PageRank. In the Section 4.4 we will describe further in detail how we could exploit the random jumps in order to personalize the search results according to the user preferences.

The artificial jumps taken by surfer can also reduce the room for spamming. A random surfer who jumps often will provide a room for the spammers to inflict the ranking. A high teleport probability means that every page is given a fixed extra “bonus” rank. Link spammers can make use of this bonus to generate local structures to inflate the importance of certain pages. Theoretically the larger values of \( \alpha \to 1 \) makes PageRank more susceptible to the small perturbation in the link structure of the webgraph. But for larger values of \( \alpha \to 1 \) the convergence of PageRank also drastically slows down [Bia05].

The random jumps induced by teleportation probability serve to be quite interesting property of PageRank. It could help the PageRank to personalize search outcomes (Section 4.4), for detection of spamming, to control the rate of convergence (depending on the values of \( \alpha \)), for stability of PageRank to perturbations in link structure, and for the design of algorithms to speed up convergence of PageRank (e.g., the second eigenvalue \( \lambda_2 \) of the transition matrix \( G \) is \( \alpha \), see Section 4.3).

### 3.3.8 The Algorithm

The Algorithm 1 will take as an input the link matrix \( A \), and therefore does the adjustments to make it stochastic and primitive. The power method applied on the modified matrix will then iteratively compute the PageRank vector \( \pi_i \). The theory of Markov chain (Ergodic Theorem) guarantees that under the conditions\footnote{Because each iteration of power method requires \( O(n) \) effort, as link matrix \( A \) is quite sparse, and on average it takes about 50 iterations to converge to the principal eigenvector of \( G \).}
Algorithm 1 The PageRank Algorithm

1: $\pi_i \leftarrow 1/n \times \text{ones}(1,n) \{\pi_i \text{ is the PageRank vector}\}$
2: \textbf{while} not converged \textbf{do}
3: $\pi_i' \leftarrow \pi_i \{\pi_i' \text{ is the PageRank vector for previous iteration}\}$
4: $\pi_i \leftarrow \alpha \pi_i A + (\alpha (\pi_i a) + (1 - \alpha)) ((1/n) \text{ones}(1,n))$
5: $\{\text{Compute convergence from } \pi_i & \pi_i' \}$
6: \textbf{end while}

described above the Algorithm 1 is guaranteed to converge to the unique and positive stationary probability distribution vector (dominant eigenvector of Markov matrix).

3.4 HITS (Hypertext Induced Topic Search)

The idea of HITS algorithm was introduced by Kleinberg in 1998\cite{Kle99}. Like PageRank, it works on the hyperlink structure of the web to calculate the relevancy scores of the webpages. Unlike PageRank, HITS first uses the conventional IR system (e.g., text-based search engine) to construct the sub-network of pages relevant to a particular query, and then computes the relevancy of the pages in the sub-network.

HITS also considers the hyperlinks between the pages as topical endorsement, i.e., a link from page $u$ devoted to topic $T$ to another page $v$ is likely to endorse the authority of $v$ with respect to topic $T$ \cite{Naj07b}. The central issue that is addressed within this framework is the distillation or characterization of broad search topic, through the discovery of “authoritative” information sources on such topics \cite{Kle99}. The focus is to extract the important information from the link structure of the web for analyzing the collection of pages relevant to a broad search topic. Hence theoretically, HITS discovers the most “authoritative” pages on those topics, with an expectation that those pages are most central pages for the broad search topics.

3.4.1 Notion of Authority

Given the subnetwork of the documents relevant to a query, how to tell that a given page is authoritative? There is no any explicitly endogenous measure of a page that would allow us to properly assess its authoritativeness \cite{Kle99}. Most of the authoritative webpages on some broad topics do not use the topic-term in their homepages. For example, for a query “search engines”, the most authoritative pages on this query could be Google, AltaVista, Yahoo!, etc., they do not use the term in their main-pages. And there is no reason to expect them to use it.

By closely analyzing the link structure sub-network of the web, we could find potential authorities through the pages that point to them. This is how we could possibly get rid of the problem identified above. The solution lies in the relationship that exists between the authorities for a topic and those pages that link to many related authorities – the latter pages are referred to as hubs \cite{Kle99}.

3.4.2 Authorities and Hubs

Like PageRank, HITS assumes each link as an implicit endorsement of the locations that it points to, that is, the links are assumed to confer authority. Unlike PageRank, HITS considers another type of pages, the hubs, which link to the popular or authoritative pages. Hence there are two types of pages in HITS terminology, authority and hub pages. HITS therefore produces two scores per page, authority score and hub score.

\footnote{In almost the same year as that of PageRank was developed (see section 3.3).}
It is observed that a certain natural type of *equilibrium* exists between hub and authority weights in the webgraph, which is exploited to develop the HITS algorithm. The algorithm iteratively identifies that equilibrium by identifying both types of the pages (authorities and hubs) simultaneously [Kle99].

### 3.4.3 Focused Subgraph

HITS operates on a *focused subgraph* (the sub-network of Web) instead of the whole webgraph, unlike PageRank. The focused subgraph can be constructed from the output of text-based search engine which can be used to iteratively produce set of pages that are most likely considered to be quite authoritative to the query topic [Kle99].

Suppose a broad-topic query, specified by a query string $q$, the subgraph corresponding to that broad-topic query will be set $Q_q$ of all pages containing the query term (results of text-based search engine). This set will be relatively large and therefore incurs considerable computational cost. And the set of desired authoritative pages may not belong to this set.

To get a better structured and computational efficient set, $k$ highest ranked pages (typically about 200 pages) will be collected for the query $q$ from the set $Q_q$. These $k$ highest ranked pages are referred to as the *Root-set* $R_q$. But still this set $R_q$ is not representative enough to help find the authoritative pages for the query topic. Because there are often extremely few links between pages in $R_q$, showing that it is essentially “structure-less”.

From $R_q$ we can now produce the set $S_q$ (see figure 3.4). The set $S_q$ is referred to as the *Base-set*. To construct the Base-set $S_q$, we seek a strong authority for the query topic, which may not be in set $R_q$, but it might be pointed to by some pages in $R_q$. Hence the set $S_q$, will be an increase in the number of strong authorities in the Rootset $R_q$.

The Base-set $S_q$ can be obtained by growing $R_q$ to include any page that pointed to by a page in $R_q$, and
any page that points to a page in $R_q$. With a restriction that each page in $R_q$ is allowed to bring in $S_q$ at most $d$ pages (typically $d = 50$) pointing to them [Kle99]. Therefore the resultant set $S_q$ will be computationally beneficial than the Rootset. Typically the Base-set contains $1000 – 5000$ nodes. We will further elaborate on the Base-set when we do experiments, see Section 5.1.1.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{connectivity.pdf}
\caption{Connectivity of Web, image taken from [Bro00]}
\end{figure}

The structure of base-set broadly resembles the overall connectivity structure of the web. The study conducted by Broder et al., [Bro00] was destined to uncover some of the intrinsic properties of the webgraph. They found a macroscopic phenomenon on the entire web, that is, the distribution of degrees on the web (in and out-degrees) are distributed according to power laws. The graph structure of the web is found to be characterized by the “bow tie” structure as shown in figure 3.5. Comparing figure 3.5 with figure 3.4, we see that broadly both of them contains three main components. It is therefore quite crucial to observe the behaviour of the Base-set in comparison to the entire webgraph. In the experiments in Chapter 5, we have observed the importance of the Base-set in identifying the set of relevant pages.

### 3.4.4 Hub and Authority scores

Authoritative pages relevant to the query should not only have high in-degree, they should also have similarities in the sets of pages pointing to them (the hubs). The authorities in the base-set are already authoritative on the common topic (content-wise); we further need to use the neighbourhoods to improve their authoritativeness. Thus pages that point to multiple relevant authoritative pages in the base-set are also important. The hub pages are therefore there to “pull together” authorities on a common topic, and allow us to disregard unrelated pages of large in-degree.
Every page therefore has double identity *hub identity* and *authority identity*. Calculating the hub and authority scores of the pages seem circular unless we characterize them as exhibiting a *mutually reinforcing relationship*: “Good authorities are pointed to by good hubs and good hubs points to good authorities” [Kle99].

The hub and authority scores can be computed iteratively for each page \( p \) in the base-set, with the help of two operations \( I \) (“in”) and \( O \) (“out”), as identified by Kleinberg.

\[
I : a_p \leftarrow \sum_{q : q \rightarrow p} h_q \tag{3.14}
\]

\[
O : h_p \leftarrow \sum_{q : q \rightarrow p} a_q \tag{3.15}
\]

\( I \) operation updates and maintains the authority scores while \( O \) operation updates and maintains the hub scores. If page \( p \) points to many pages with large hub scores than it should receive a large authority score and vice versa, which signifies the numerical interpretation of the mutually reinforcing relationship.

The operations shown in equations (3.14) and (3.15), will iteratively filter out the top authorities and hubs, and converge from the sequence of vectors \( \vec{a}_k \) and \( \vec{h}_k \) to the fixed points \( \vec{a}^* \) and \( \vec{h}^* \).

### 3.4.5 Principal Eigenvectors

The operations \( I \) and \( O \) in equations (3.14) and (3.15) can be represented in matrix notation with the help of an adjacency matrix \( A \) such that:

\[
A_{ij} = \begin{cases} 
1 & \text{if there is a link from page } P_i \text{ to } P_j \\
0 & \text{otherwise.}
\end{cases}
\]
For the figure 3.2 the corresponding adjacency matrix $A$ can be:

$$
A = \begin{pmatrix}
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0
\end{pmatrix}
$$

Thus $I$ and $O$ in terms of the adjacency matrix can be:

$$
a = A^T h \\
h = A a
$$

and also we could further write the above two equations as:

$$
a = A^T A a \\
h = A A^T h
$$

The matrices $A^T A$ and $A A^T$ are symmetric, positive semidefinite and nonnegative matrices [Gol96; Lan06]. In Linear Algebra the above two equations correspond to the standard problem of finding the dominant eigenvectors of the matrices $A^T A$ and $A A^T$ respectively. A standard result in linear algebra [Lay94; Gol96] states that if $M$ is a symmetric matrix and $v$ is a vector not orthogonal to the principal or dominant eigenvector of $M$, then the unit vector in the direction of $M^k v$ converges to the principal eigenvector of $M$, as $k \to \infty$. And if $M$ is nonnegative the principal eigenvector of $M$ will also be nonnegative. Thus the problem of finding the authority and hub score reduces to the standard problem of finding the principal eigenvectors of matrices $A^T A$ and $A A^T$ respectively, thus:

$$
a^k = A^T A a^{k-1} \quad (3.16) \\
h^k = A A^T h^{k-1} \quad (3.17)
$$

$a^*$ will be the principal eigenvector of $A^T A$ and $h^*$ the principal eigenvector of $A A^T$ as $k \to \infty$ [Kle99]. Thus the symmetric, semidefinite and nonnegative matrices $A^T A$ and $A A^T$ assure the convergence of HITS algorithm within finite number of iterations of the Power method. The rate of convergence is given by the rate at which $(\frac{1}{\lambda_{\text{max}}})^k \to 0$ as $k \to \infty$ [Lay94]. For more information on the convergence behaviour, see Section 4.3.2.

Theoretically the convergence of the iterative problem in equations (3.16) and (3.17) is quite rapid, according to [Kle99] one can essentially reach the convergence within $k = 20$. But in the practice the convergence behaviour might vary, from somewhere around $k \in (20 - 160)$ iterations, see for example the results in Table 5.3.

Instead of iterating until convergence, one could have a bounded iterative process, by restricting iterations until some threshold $p$ (number of iterations) and therefore compute scores $a^p$ and $h^p$ instead of $a^*$ and $h^*$.

The process of finding the principal eigenvectors, equations (3.16) and (3.17), emphasizes the underlying motivation to reinforce the relations $I$ and $O$ in equations (3.15).

### 3.4.6 Non-Unique authority or hub score and Adjustment

Despite the nice formulation of the $I$ and $O$ operations and its capability to converge rapidly, there is still one another concern that need to be addressed; that is, the issue of uniqueness of authority and hub vectors.

Considering the structure of the adjacency matrix $A$, it might have multiple principal eigenvalues i.e., repeated roots of the characteristic polynomial, $\det(A^T A - \lambda I)$. Therefore there will be multiple principal eigenvectors

---

5Because they resemble the standard equation for finding the eigenvalue and eigenvector of a matrix $A$ i.e. $\lambda x = Ax$
(authority and hub vectors) corresponding to the repeated principal eigenvalues. If we observe more closely the uniqueness problem is actually due to the reducibility of the matrix $A$ [Gol96]. This, in terms of the graph theory means that there are set of states that it’s possible to enter, but once entered it’s not possible to leave (multiple components in the graph). And that is similar to the problem identified by Page and Brin (see Section 3.3), in the PageRank algorithm.

Irreducible matrices are those in which every state is reachable from every other state. Equivalently the matrix $A$ is irreducible if the graph $G$ is strongly connected. And according to the Perron-Frobenius theorem [Kit98; Bré99; Lay94], an irreducible nonnegative matrix possesses a unique and positive principal eigenvector. A same kind of adjustment as in PageRank, can be applied to the HITS as well, i.e., modify the matrices $A^T A$ and $A A^T$, to:

$$\sigma A^T A + \frac{(1-\sigma)}{n}ee^T$$

$$\sigma A A^T + \frac{(1-\sigma)}{n}ee^T$$

This will ensure that the matrices $A^T A$ and $A A^T$ will be primitive and therefore irreducible, based on the same reasons as in PageRank. For more detailed discussion on modifications of HITS, see Section 4.4.

3.4.7 Random Walks and HITS

The natural question to ask here is that whether the output of HITS algorithm can be seen as the stationary distribution of certain random walk(s) on the underlying graph (base-set), like PageRank. This has been shown by Borodin et al., [Bor01] that indeed the output of HITS algorithm can also be seen as the stationary distribution of random walk. We present the theorem here without proof.

**Theorem 1** There exist sequences $M_1^a, M_2^a, \ldots, M_n^a, \ldots$, and $M_1^h, M_2^h, \ldots, M_n^h, \ldots$ of Markov Chains, such that, for each $n \geq 1$, the stationary distribution of $M_n^a$ is equal to the authority vector after the $n^{th}$ iteration of Kleinberg’s algorithm, and the stationary distribution of $M_n^h$ is equal to the hub vector after the $n^{th}$ iteration of Kleinberg’s algorithm.

According to the fundamental theorem of Markov chains for random walks [Mot95], the stationary distribution of the Markov chain $M_n^a$, is the same as the authority vector $\vec{a}$ of HITS algorithm. And similarly the hub vector $\vec{h}$ corresponds to the stationary distribution $M_n^h$. For the proof of the theorem 1 see [Bor01]. Hence the rankings produced by HITS algorithm can be seen as the stationary distribution of two random walks on Base-set.

3.4.8 Singular Value Decomposition

The $I$ and $O$ operations can also be viewed in terms of the important factorization technique in linear algebra, the Singular Value Decomposition (SVD). SVD is a powerful technique for data analysis. It reveals a great deal about the structure of the matrix. It is usually applied as a dimensionality reduction tool, to obtain a concise and compact representation of a large datasets, mostly applied in image and signal processing and essentially in multimedia IR models. It has also a large scale application in matching the large datasets with each other in IR. In Section 2.6.2, Latent Semantic Indexing model also employed SVD to reduce the dimensionality of the term matrix. Hence it is crucial to have an overview of the use of SVD in LAR models, specifically HITS.
In case of HITS, authority and hub vectors actually represent the singular vectors of the adjacency matrix $A$ [Lay94].

$$A = \mathbf{U}\Sigma\mathbf{V}^T = [\vec{u}_1 \quad \vec{u}_2 \quad \ldots \quad \vec{u}_r]$$

And the outer product form can be written as:

$$A = \sigma_1 \vec{u}_1\vec{v}_1^T + \sigma_2 \vec{u}_2\vec{v}_2^T + \ldots + \sigma_r \vec{u}_r\vec{v}_r^T$$

The matrices $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices\(^6\). The diagonal matrix $\Sigma$ contains $r$ non-zero singular values ($\sigma_1, \sigma_2, \ldots, \sigma_r$) which are the square roots of eigenvalues of $A^TA$ and $A^T A$ [Lay94; Gol96]. The vectors corresponding to the columns of matrix $\mathbf{U}$ ($\vec{u}_i$, for $i = 1, 2, \ldots, r$) represent the left singular vectors of matrix $A$, and they are the eigenvectors of $A^T A$. While vectors corresponding to the rows of matrix $\mathbf{V}^T$ ($\vec{v}_i$, for $i = 1, 2, \ldots, r$) form the right singular vectors of matrix $A$ and the eigenvectors of $A^T A$. From the above full form of SVD we will have:

$$A = \mathbf{U}\Sigma\mathbf{V}^T$$

$$A^T = \mathbf{V}\Sigma^T\mathbf{U}^T$$

$$A^T A \vec{u}_i = \sigma_i^2 \vec{u}_i \quad \text{for} \quad i = 1, 2, \ldots, r \quad (3.19)$$

$$A^T A \vec{v}_i = \sigma_i^2 \vec{v}_i \quad \text{for} \quad i = 1, 2, \ldots, r \quad (3.20)$$

As the columns of $\mathbf{U}$ and rows of $\mathbf{V}^T$ are orthogonal (because they are eigenvectors of $A^T A$ and $A^T A$ respectively), therefore any $k$ of them define a basis for $k$–dimensional space. This leads to the observation in the next section.

**Linear Trend**

It should come as no surprise that extracting the principal eigenvector (component) of matrix $A^T A$ amount to a form of linear trend or factor analysis. Each principal component of $A^T A$ represents an orthogonal direction in which there exist highly-correlated pages. The tightly knit communities therefore correspond exactly to the orthogonal factors of the link matrix $A$. And hence the most heavily-linked of these tightly knit communities emerge as the principal component or the authoritative sources.

From equations (3.19) and (3.20) we have:

$$A \mathbf{V} = \mathbf{U}\Sigma$$

$$A^T \mathbf{U} = \mathbf{V}\Sigma$$

To describe the mathematical expressions in equations (3.21) and (3.22), the matrix $A$ can be thought of as a matrix that associates two different types of entities: objects (rows) with attributes (columns). Objects

\(^6\)Over the complex field the unitary matrices correspond to the orthogonal matrices. In particular $\mathbf{U} \in C^{m \times n}$ is unitary if $\mathbf{U}^T \mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_n$ [Lay94; Gol96]
are expressed as vectors in attribute space and attributes as vectors in the object space. In equation (3.21) the product $\mathbf{AV}$ defines projection of objects (row vectors) from the attribute space on the feature space defined by the column vectors of matrix $\mathbf{V}$. The matrix $\mathbf{V}$ defines the directions on which the object vectors are projected to, and $\mathbf{U}\Sigma$ defines the mapping of the object vectors to the feature space. And similarly the opposite (see equation (3.22)), when $\mathbf{U}$ defines the direction and $\mathbf{V}\Sigma$ defines the position of projection. See figure 3.7 for the graphical representation. In linear algebra, this corresponds to the fact that the columns of matrix $\mathbf{U}$ represent the orthonormal basis for $\text{span}(\mathbf{A})^7$. Similarly equation (3.22) shows that matrix $\mathbf{V}$ correspond to the orthonormal basis for $\text{span}(\mathbf{A}^T)$.

From the observations above and equations (3.19) and (3.20), in HITS, the hub vector $\vec{h}$ and the authority vector $\vec{a}$ correspond to the right and left principal singular vectors of $\mathbf{A}^T$ respectively. In the matrix $\mathbf{A}^T$ the $\text{span}(\mathbf{A}^T)$ correspond to the hub space. The $\vec{h}$ vector captures the strongest trend within hub space. The authority vectors are projected on these vectors. The projection lengths are the authority weights and they capture how closely each authority is aligned with the strongest trend $\vec{h}$. Thus authority vector $\vec{a}$ is also the strongest linear trend in the hub space. Similarly in the matrix $\mathbf{A}$ the $\text{span}(\mathbf{A})$ correspond to the authority space. With the same observation the hub vector $\vec{h}$ is also the strongest linear trend in the authority space.

A vector $v$ is linear trend in matrix $\mathbf{A}$, if it shows the tendency of the row of vectors of $\mathbf{A}$ to align with $v$. In fact $\vec{u}_i$ and $\vec{v}_i$ are the $i^{th}$ strongest linear trend, and $\sigma_i$ the strength of $i^{th}$ strongest linear trend [Tsa04a]. Hence SVD is used in IR to discover the linear trends in the data. Relating figure 3.7 and the description above with HITS; it actually tends to discover the strongest linear trend in the authority space.

3.4.9 TKC Effect

As described above, we try to map the authority and hub vectors to the hub and vector space of spanned by column and rows of matrix $\mathbf{A}$ respectively. An intriguing question is to understand how the structure of the
graph affects the ranking behaviour of the algorithm. And considering the fact that HITS computes a densely linked collection of pages without regard to their content [Kle99], it is expected to favour pages that belong to most dense component of the graph (either in authority or hub). According to [Bor05; Lem00] in a graph containing multiple communities (densely connected components), the HITS algorithm will only focus on one of them in the top positions of the ranking, the one that contains the hubs and authorities that are most tightly interconnected. This phenomenon is therefore termed as Tightly Knit Community (TKC) effect [Bor05].

It has been analyzed and stated that HITS favour TKC effect [Kle99; Cha99; Dri99]. In fact the TKC effect is the by-product of the properties of Singular Value Decomposition, as we described in the previous section. The tendency of SVD to discover the linear trends in data, and HITS likes the strongest linear trend. Therefore, from the figure 3.7, the resultant authority and hub vectors tend to be in the densest component of the graph. Thus the tightly interconnected set of nodes in the graph try to attract HITS algorithm, and HITS gets attracted, therefore possibly causes “topic drift”\(^8\).

\[\text{3.4.10 Algorithm} \]

\begin{algorithm}
1: \( \mathbf{A} \) : adjacency matrix formed from the base-set \( S_q \)
2: \( a^{(0)} \) : set the seed values of the authority vector
3: \( h^{(0)} \) : set the seed values of the hub vector
4: \textbf{while} not converged \textbf{do}
5: \( A^k \leftarrow \mathbf{A}^T \mathbf{A} a^{k-1} \)
6: \( \mathbf{h}^k \leftarrow \mathbf{A}^T \mathbf{h}^{k-1} \)
7: \( a^k \leftarrow a^k \) Normalize
8: \( \mathbf{h}^k \leftarrow \mathbf{h}^k \) Normalize
9: \( k \leftarrow k + 1 \)
10: \{ Compute the convergence \}
11: \textbf{end while}
\end{algorithm}

Like PageRank algorithm, in the Algorithm 2 we need the adjacency matrix \( \mathbf{A} \) as an input. There is another algorithm employed to construct the base-set \( S_q \), corresponding to a query \( q \), from root-set \( R_q \), and from the set \( Q_q \), see [Kle99]. Constructing the adjacency matrix \( \mathbf{A} \) from the base-set \( S_q \) will then be straight forward. Also note that the above iterative method is the famous Power method as used in PageRank. It also requires the starting or seed vectors \( a^{(0)} \) and \( h^{(0)} \), a general rule is that it should be in the range of the characteristic polynomial i.e., \( \text{det}(\mathbf{A}^T \mathbf{A} - \lambda I) \). By the construction of the algorithm (the power method), any arbitrarily chosen non-negative seed vectors \( a^{(0)} \) and \( h^{(0)} \) (such as \( h^{(0)} = 1/\text{ne} \)) also convergences to an eigenvector of the largest eigenvalue [Gol96].

In the Chapter 4 we will further analyze in detail the intriguing properties of HITS and the possibility to improve and / or adjust some of the limitations.

3.5 SALSA (Stochastic Approach for Link-Structure Analysis)

SALSA, the stochastic approach towards LAR also exploits mainly the link-structure of the web, was developed by Lempel and Moran, 2000. SALSA makes use of the Kleinberg’s mutual reinforcement approach and the ran-
dom walk of the Markov chain theory as is used in PageRank. The initial assertion was to employ an algorithm which would computationally be more efficient than the mutual reinforcement approach of Kleinberg [Lem00].

Like HITS, SALSA creates both hub and authority scores for each document on the Web, and like PageRank, the scores are calculated using the theory of Markov chains [Lan06]. However, in contrary to HITS’s mutual reinforcement approach, SALSA compute hub and authority scores on the neighbourhood graph (focused subgraph) by performing two independent random walks.

### 3.5.1 Informative and Non-informative links

Considering the link between pages p and q, (p → q), p implicitly suggests, or recommends that the page q also contains a relevant topic of interest as in p. Such type of link is called informative link. In the wake of Sections 3.3 and 3.4, page p endorses page q, and therefore the page q is considered important and hence a candidate to be visited by a surfer visiting page p.

But all the links cannot be categorized as informative links; unfortunately there are many links which confer little or no authority. For example, intra domain links or navigational links, commercial or sponsor links, and links which result from link-exchange agreements [Lem00]. A crucial task prior to applying any link analysis based ranking is to identify the informative and non-informative links, and therefore prune the non-informative links from the graph. These pre-processing to the graph create a much realistic link structure between the pages, which could be used to compute a more relevant and pragmatic set of authorities.

### 3.5.2 Bipartite Graph (Hubs and Authorities)

In the stochastic approach (SALSA), the coupling between the hubs and authorities are less tight then HITS. It actually considers an undirected bipartite graph $G^9$, whose two parts correspond to hubs and authorities respectively (see figure 3.8). An edge between the hub $P_i$ and the authority $P_j$ means that there is an informative link from $P_i$ to $P_j$. And therefore pages pertaining to dominant topics in $G$ should be highly reachable from many pages (nodes in the bipartite graph).

Considering the graph of five pages in figure 3.2, the example graph in figure 3.8 depicts the hubs and authorities corresponding to that graph.

In the stochastic approach of analyzing the link structure, the theory of random walks of Markov chain are combined with the notion of hubs and authorities, the two sides of the bipartite graph (figure 3.8). Therefore the algorithm conducts two different and independent random walks corresponding to the hubs and authorities respectively. But unlike the conventional random walks, state transitions in these chains are generated by traversing one link forward and one link backward [Lem00].

The undirected graph $G = (V_h, V_a, E)$ (e.g., the figure 3.8), can be constructed from the site-collection (the base-set $S_q$, given a query $q$, see Section 3.4). We call the base-set here as the site-collection $C$.

$$
V_h = \{ s_h \mid s \in C \text{ and } \text{out-degree}(s) > 0 \} \text{ (hub side of } G) \\
V_a = \{ s_a \mid s \in C \text{ and } \text{in-degree}(s) > 0 \} \text{ (the authority side of } G) \\
E = \{ (s_h, r_a) \mid s \rightarrow r \in C \} \text{ (the set of directed edges in } C) 
$$

From above formulation, in our example graph in figure 3.8 we have:

$$
V_h = \{ P_3, P_1, P_4, P_5, P_2 \} \\
V_a = \{ P_2, P_3, P_1, P_4, P_5 \}
$$

$^9$The graph $G$ can be constructed from the neighbourhood graph made in similar way as in HITS, focused subgraph (see Section 3.4.3)
Recall the adjacency matrix $A$ of figure 3.2, which can be used in this case too:

$$A = \begin{pmatrix}
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0
\end{pmatrix}$$

And now assume two new matrices $A_r$ and $A_c$ derived from matrix $A$ by normalizing it’s entries in some way. Matrix $A_r$ is formed by normalizing each row of $A$ to sum to 1. Similarly $A_c$ is formed by normalizing the entries of $A$ such that for each column the sum of entries is 1. For our example figure 3.2 we have:

$$A_r = \begin{pmatrix}
0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0
\end{pmatrix}, \quad A_c = \begin{pmatrix}
0 & 1/3 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1/3 & 0 & 0 & 0 \\
1/2 & 1/3 & 1/2 & 0 & 0 \\
1/2 & 0 & 0 & 1 & 0
\end{pmatrix}$$

### 3.5.3 Two Random Walks

On the bipartite graph in figure 3.8, two distinct random walks are conducted. Each walk is meant to traverse the pages on each one of the two sides.

For the first walk, random surfer starts off uniformly at random from any node on authority side. The walk will then proceed alternating between backward and forward steps. When the surfer is at the authority side of the graph, one of the incoming links is selected uniformly at random and move to the hub side of the graph.
Similarly when at the hub side of the graph, one of the outgoing links is selected uniformly at random and move to the authority side of the graph. And hence the authority weights vector will be the stationary distribution of this random walk, starting from authority side. This first random walk will be conducted on the set $V_a$ (the set of authority nodes), described above.

The second walk will be conducted on the set $V_h$. Here the surfer starts off from the any node on hub side. And proceeds by alternating between forward and backward steps. When the surfer is at hub side it will select one of the outgoing links uniformly at random and move to authority side and vice versa. The hub weights vector like authority vector will be the stationary distribution for this second random walk, starting from hub side.

With this settings, we may expect to have $t$-authorities relevant to topic $t$ amongst the nodes most frequently visited by the random walk on $V_a$, and similarly we expect to have $t$-hubs will nodes amongst the most frequently visited by the second random walk on $V_h$ nodes.

3.5.4 Stochastic Matrices

The transition matrices of the two Markov chains representing the random walks on hub and authority side of the bipartite graph can be defined mathematically here.

Refer to the walk starting off from authority side of graph conducted on the set $V_a$, the Markov chain of this random walk has transition probabilities:

$$\tilde{a}_{ij} = \sum_{\{k \mid (i_h, k_a), (j_h, k_a) \in G\}} \frac{1}{\deg(i_h)} \cdot \frac{1}{\deg(k_a)}$$

In terms of backward and forward hops of random surfer we can mathematically have:

$$\tilde{a}_{ij} = \sum_{\{k \mid k \in B(i) \cap B(j)\}} \frac{1}{|B(i)|} \cdot \frac{1}{|F(k)|}$$

For some node $i$ we have $B(i) = \{ j : A(j, i) = 1 \}$, i.e., $B(i)$ is the set of back-links for node $i$. And $F(i) = \{ j : A(i, j) = 1 \}$, is the set of forward-link for node $i$.

The random walk conducted on the authority graph\(^{10}\) will have a probability $\tilde{a}_{ij}$ when the surfer moves from authority $i$ to $j$.

Similarly for the random walk that starts from hub side, the transition probabilities are:

$$\tilde{h}_{ij} = \sum_{\{k \mid (i_h, k_a), (j_h, k_a) \in G\}} \frac{1}{\deg(i_h)} \cdot \frac{1}{\deg(k_a)}$$

And simplifying to forward and backward hops of random surfer, we have:

$$\tilde{h}_{ij} = \sum_{\{k \mid k \in F(i) \cap F(j)\}} \frac{1}{|F(i)|} \cdot \frac{1}{|B(k)|}$$

Same interpretation of random walk on hub graph\(^{11}\).

A positive transition probability $h_{ij} > 0$ implies that a certain page $k$ is pointed to by both the pages $i$ and $j$, and hence page $j$ is reachable from page $i$ by two steps: forward-link on $i \rightarrow k$ and then back-link from

\(^{10}\)When authority $i$ and $j$ share a hub, we place an undirected edge between authority $i$ and $j$, the subsequent graph is the authority graph.

\(^{11}\)When hub $i$ and $j$ share an authority, we place an undirected edge between hub $i$ and $j$, the underlying graph will be hub graph.
3.5 SALSA (STOCHASTIC APPROACH FOR LINK-STRUCTURE ANALYSIS)

Figure 3.9: Hub and Authority graph respectively

$k \rightarrow j$ [Lem00]. See figure 3.9(a). Similarly $a_{ij} > 0$ implies that a certain page $k$ points to both pages $i$ and $j$ ($k$ is their common authority). Thus $i$ is reachable from $j$ by two steps: back-link from $j \rightarrow k$ and then forward-link on $k \rightarrow i$. See figure 3.9(b).

Consider now the matrices $A_r$ and $A_c$, the stationary distribution vector $a$ of Markov chain corresponding to random walk on authority graph is actually the principal right eigenvector of matrix $A_c^T A_r$ i.e.,:

$$a = A_c^T A_r a$$

where the vector $a$ is a column vector.

And the stationary distribution vector $h$ (a column vector) of the Markov chain on hub graph is the principal right eigenvector of matrix $A_r A_c^T$ i.e.,:

$$h = A_r A_c^T h$$

SALSA can also be interpreted in terms of the operations in HITS. As in HITS, the $I$ operation, hubs broadcast their weights to the authorities, and authorities sum up the weights of hubs that point to them. SALSA instead of broadcasting, in $I$ operation, each hub divides its weight equally among its’ out-links (the authorities that it points to). And in $O$ operation of SALSA algorithm each authority equally divides it’s weight among its’ back-links (the hubs). Thus the modified operations in SALSA are now:

$$I : a_p \leftarrow \sum_{(q \in B(p))} \frac{1}{|F(q)|} h_q$$

$$O : h_p \leftarrow \sum_{(q \in F(p))} \frac{1}{|B(q)|} a_q$$

Lempel considers the matrix $\tilde{A}$ to have the non-zero rows and columns of $A_r A_c^T$ and the matrix $\tilde{H}$ to have the non-zero rows and columns of $A_c^T A_r^T$. The graph $G$ is assumed to be connected, which therefore causes the matrices $\tilde{A}$ and $\tilde{H}$ to be irreducible.

The stochastic matrices $\tilde{A}$ and $\tilde{H}$ are therefore primitive, because the Markov chains which they represent are aperiodic [Lem00]. Therefore every node on either side of the bipartite graph has a self loop, causing the chain to be aperiodic. The matrices are also symmetric, considering matrix $\tilde{A}$ , if $\tilde{a}_{ij} > 0$ then $\tilde{a}_{ji} > 0$, furthermore $a_{ij} > 0$ if and only if $[A^T A]_{ij} > 0$. Similarly for matrix $\tilde{H}$, if $\tilde{h}_{ij} > 0$ then $\tilde{h}_{ji} > 0$. Hence $\tilde{h}_{ij} > 0$
if and only if $[\mathbf{A} \mathbf{A}^T]_{ij} > 0$. Where matrix $\mathbf{A}$ is the original adjacency matrix corresponding the site-collection $\mathbf{C}$ or the base-set.

Like PageRank and HITS, the problem of reducibility can cause problems in SALSA too. This means that it is possible to have authority graph which is not connected and similarly the same could happen with hub graph. Or in general the graph $\mathbf{G}$ can be not-connected (it may have disconnected components). Under such circumstances simply multiply each page’s authority score by the normalized size of the irreducible component to which it belongs, for details see [Lem00].

But when the graph $\mathbf{G}$ is connected i.e., it contains a single component, the stochastic authority(hub) weights will be proportional to the in(out) degree. SALSA therefore reduces to primitive algorithm, the InDegree, when $\mathbf{G}$ has a single connected component only.

### 3.5.5 Algorithm

**Algorithm 3 The SALSA Algorithm**

1: $\mathbf{A} :$ adjacency matrix formed from the base-set $S_q$
2: $\mathbf{A}_c :$ matrix $\mathbf{A}$ where columns normalized to sum to 1
3: $\mathbf{A}_r :$ matrix $\mathbf{A}$ where rows normalized to sum to 1
4: $\tilde{\mathbf{A}} :$ non-zero rows and columns of $\mathbf{A}_c^T \mathbf{A}_r$
5: $\tilde{\mathbf{H}} :$ non-zero rows and columns of $\mathbf{A}_r \mathbf{A}_c^T$
6: $a^0 :$ set the initial value of the authority vector
7: $h^0 :$ set the initial value of the hub vector
8: while not converged do
9:   $\mathbb{I} : a^k \leftarrow \tilde{\mathbf{A}} a^{k-1}$
10: $\mathbb{O} : h^k \leftarrow \tilde{\mathbf{H}} h^{k-1}$
11:  $a^{k'} \leftarrow a^k$ Normalize
12:  $h^{k'} \leftarrow h^k$ Normalize
13:  $k \leftarrow k + 1$
14:  {Compute the convergence}
15: end while

The Algorithm 3 reflects all the findings that we had in the previous sections. We have experimented with the Algorithm 3 in Chapter 5. SALSA is considered to be less vulnerable to TKC effect (described in Section 3.4.9). Najork [Naj07a] had conducted experiments on large Webgraph to evaluate HITS and SALSA. He found that SALSA “substantially outperformed” HITS. The results that we have found in our study are a bit different, in our own specific settings. See Appendices A and B for comparison.
Part II

Evaluations, Analyses and Experiments
Chapter 4

Evaluations and Analyses of LAR Models

4.1 Link Analysis Ranking Process

The presence of (hyper-) link information clearly augmented a great deal to the characterization of the informative content present in the documents. LAR approaches are intended to resolve some of the intrinsic weaknesses of the content based IR models. Through the analysis of network or web structure of the documents (due to citation structure) LAR approaches bring in a whole new horizon to IR space.

The essence of LAR therefore is that the “overall information” of a hyperlink database of documents is not composed of only static “textual information”, but also another, the “hyper” information.

The LAR models rely on the documents database with index structure having entries equipped with hyperlink information. The first step therefore towards analyses of the LAR algorithms is to have a hyperlinked collection of documents. From the hyperlinked information stored in index structure we create a hyperlink or adjacency graph, similar to the one described in last chapter, figure 3.2. In this graph each document is represented as a node and each hyperlink is represented as a directed edge between the nodes (documents) similar to the webgraph description in Section 3.3.

Usually in most of the LAR models, the edges between the documents are un-weighted, i.e., the costs of transition between the pages are assumed to be uniform. But there are models that work on weighted graphs, the study conducted by Bharat and Henzinger [Bha98] concerns with weighted graph by using measures from content analysis of the documents, together with LAR. In another study by Cohn and Hofmann [Coh01], a joint probabilistic model has been introduced for modelling the contents and hyperlinks. Hence probabilistic theories are used to make predictions about the existence and strengths of hyperlinks and citations.

The graph can be generated for the entire collection of documents independent of any query (query independent) or it can be generated from a subset of the documents collection based on a given query (query dependent). PageRank as discussed in Section 3.3, works on whole webgraph irrespective to any query. While HITS and SALSA in Sections 3.4 and 3.5 work on a smaller or focused subgraph based on a given query. This idea of query (in)dependence correspond to the initial perspectives of IR models (see descriptively in Section 2.2 and graphically in figure 2.1).

The resultant hyperlinked graph (query dependent or independent) will be given as an input to the LAR algorithms. This graph will be encoded in an adjacency matrix $A$, where $A[i, j] = 1$ if there is a link from node $i$ to node $j$ and 0 otherwise (see also Section 3.3.3). The LAR algorithm iteratively operate on the hyperlinked graph (the adjacency matrix $A$) and returns the rank vector ($n$-dimensional) $\vec{x}$ with weights computed for each
node in the graph, where \( x_i \) is the weight of \( i^{th} \) node in the graph. The weights are actually the \textit{probabilities of relevance} of each document to the user query. They are therefore used to rank the retrieved documents (sort by weights). LAR algorithms are intended to discover ranking of documents which maximize the relevance of user query to the ranked and retrieved documents. LAR algorithms are thus meant to discover \textit{authoritative} documents through analyzing the hyperlink graph \[Tsa04a\].

Many interesting ideas have been brought in LAR space over the years. Starting from \textit{Bibliometrics} [Lar96], then moving towards making more active use of “hyper” information together with “textual” information [Mar97], and then exclusively using only hyperlink information with objectives to maximize relevancy between user query and search outcomes. The application of core concepts from linear algebra, such as; the Markov chain theory (as described in Chapter 3), LAR as a system of linear equations, the use of Singular Value Decomposition, Dynamic programming, Extrapolation techniques, Optimization algorithms (such as Gradient Ascent), and many more intriguing ideas have been investigated and incorporated in LAR.

In the Chapter 3 we have reviewed the existing research in LAR. There were broadly two families of LAR algorithms, \textit{query-independent} algorithms and \textit{query-dependent} algorithms. In the former case, the algorithm ranks the whole document corpus independent of any query. While the latter ranks a subset of the document corpus depending on some specific query. The two pioneer algorithms PageRank and HITS are query-independent and query-dependent respectively. They were followed by substantial amount of research [Bri06; Hav03a; Jeh03; Kam03b; Dri99], to name just a few. IR practitioners now have a wider range of LAR algorithms to choose from.

In this chapter our focus is to purposely explore the intrinsic and implicit properties of query-dependent LAR algorithms. We will also study some of interesting properties and characteristics of query-independent algorithms and try to incorporate them in the query-dependent ones. We will go through different improvements (both in query-dependent and independent) and try to relate them to inherent properties of query-dependent algorithms particularly HITS and SALSA. We will also study the behaviours of different algorithms. The ideas discussed in this chapter and Chapter 3 will be supplemented by the empirical analyses, in Chapter 5.

\subsection*{4.2 Implicit Properties of HITS algorithm and Problems}

HITS algorithm can be analyzed from its inherent idea, i.e., the \textit{mutually reinforcing relationship} between hubs and authorities. “A good authority is the one that is pointed by many good hubs and a good hub is the one that points to many good authorities” [Kle99]. Thus, hubs and authorities mutually strengthen each other. The quality of hub depends on the quality of the pages pointed to by this hub (the authorities) and the quality of authority depends on the quality of pages pointed to this authority (the hubs). Hub and Authority weight are calculated in a cyclic fashion; hub weights are calculated from authorities and authority weights are calculated from hubs.

The association described above between hub and authority weights is captured through the \textit{addition operation} [Kle99]. Hub weight of a page is calculated as the sum of the authority weights of the pages that pointed to by this hub, similarly authority weight of a page is calculated as the sum of the hub weights of the pages that point to this authority node. As identified by [Tsa04a], the definition of mutually reinforcing relationship has \textit{two implicit properties}. First that it is \textit{symmetric}, because both hub and authority weights are calculated in same way. This means that if the orientation of graph is reversed then authority and hub weights of original graph will be hub and authority weights of reversed graph respectively. Secondly that it is \textit{egalitarian}, while computing the authority weight of a page, the hub weights of the pages that point to this page are all treated
equally. Similarly, when calculating the hub weight of a page, the authority weights of the pages that it points to are considered equally. We will further elaborate these two properties with a help of example graphs.

### 4.2.1 Example

![Graph](image)

(a) Common Authority  
(b) Common Hub

**Figure 4.1: Problems in HITS.**

In figure 4.1, we see two components, the one in figure 4.1(a) with a central authority and the other one in figure 4.1(b) with a central hub. In the red component (figure 4.1(a)) many hubs points to one authority while in the blue component (figure 4.1(b)) one hub points to many authorities. If the blue hub points to more authorities than the number of red hubs, HITS will allocate all authority weight to the blue authorities and giving zero weight to the red authority. It is because the blue hub is considered as best hub and therefore the blue authorities receive more weight than the red authority.

Intuitively the red authority seems to be better than blue authorities and should have been ranked higher. But HITS produced a non-intuitive result and therefore ranked the blue authorities higher than the red one. This means that by linking to a lot of pages it will end up in a non-intuitive result. And therefore HITS is susceptible to link spamming.

Symmetry relationship between hubs and authorities treats both hub and authority the same. A node with high in-degree is likely to be a good authority, but a node with high out-degree should not necessarily be a good hub. Because with such assumption there are high possibilities of spamming. In such a situation it would be very easy to inflict ranking by artificially uplifting the hub weights by adding links to a lot of pages. In the Section 4.2.2 we will present some approaches to cope up with this problem, and some more problems identified there.

There are similar kind of problems identified by [Bha98]. Three problems are identified in connectivity analyses of HITS.

Firstly the mutually reinforcing relationships between hosts; certain arrangements of pages conspire to dominate and inflict the computation of ranking. For example, a set of pages on one host point to a single page on a second host, same as the red component in the figure 4.1(a). This drives up the hub scores of the pages on the first host and the authority score of the page on the second host [Bha98]. The reverse case is similar to the blue component in the figure 4.1(b). Secondly the automatically generated links; to insert links automatically or spam links in order to conspire the ranking. These two problems are the same as described in the figure 4.1.

Thirdly the non-relevant documents; it is highly probable that the focused subgraph (the base-set) contain
non-relevant nodes to the query topic. If the non-relevant nodes are well connected then the topic drift problem can arise, which means that the most highly ranked authorities and hubs will not be relevant to query topic. The consequences of the tightly connected non-relevant documents on ranking are exaggerated by the TKC effect of HITS, see Section 3.4.9.

4.2.2 Approaches to address the problems

The problems identified in the last section have significant side-effects on HITS algorithm. The example showed that blue hub (figure 4.1(b)) can inflict the hub and authority weights by linking to a lot of pages. The blue hub will have more score if it points to a lot of authorities irrespective of its authoritativeness. And similarly the blue authorities will get more score from a hub pointing to a lot of authorities. This means that there are definite loopholes for the spammers, to illicitly use link spamming to conspire the ranking.

Hub Average

As proposed by [Tsa04a] a modification to HITS algorithm could possibly solve the problems identified in figure 4.1. There is one natural modification needed in $\odot$ operation, i.e., the hub weight of a node $i$ is set to be the average authority weights of the authorities pointed by $i$. Hence,

$$a_i = \sum_{j \in B(i)} h_j \quad \text{and} \quad h_i = \frac{1}{|F(i)|} \sum_{j \in F(i)} a_j$$  \hspace{1cm} (4.1)

where $B(i)$ is the set of back-links of node $i$, while $F(i)$ is the set of forward-links for node $i$. $|F(i)|$ is cardinality or the number of the forward-links.

This modification was originally presented by [Bor01]. The resultant algorithm is called as HubAvg (Hub Average). The intuition is that a good hub should point only (or at least mainly) to good authorities [Tsa04a]. In the example identified in the last section HubAvg assigns the same weights to both the red and blue hubs, and it identifies the red authority as the better authority than the blue ones.

HubAvg can be seen as ‘hybrid’ of HITS and SALSA algorithm [Bor01] (see Section 3.5). The $\odot$ operation is the same as HITS, i.e., broadcasting the authority weight to hubs, and $\mathbb{I}$ operation is like SALSA, i.e., dividing the hub score to the authorities. If the same modification in $\odot$ operation can be applied to $\mathbb{I}$ operation too, i.e., both hub and authority nodes divides their weights (instead of broadcasting) to the subsequent nodes, then HITS algorithm will reduce to SALSA algorithm, as described in Section 3.5.4.

Expressing these changes in matrix notation, we have:

$$A_r = FA$$  \hspace{1cm} (4.2)

where $A$ is the adjacency matrix and $F$ is a diagonal matrix with $F(i,i) = 1/|F(i)|$. While $F(i)$ is the set of forward-links as defined before. Thus we have:

$$M_{HA} = A^T A_r$$  \hspace{1cm} (4.3)

$$M_{HA} = (F^\frac{1}{2} A)^T (F^\frac{1}{2} A)$$  \hspace{1cm} (4.4)

The modified authority weight will be the principal right eigenvector of the matrix $M_{HA}$. The power iterations for authority vector $a$ will be:

$$a_k = M_{HA} a_k^{k-1}$$  \hspace{1cm} (4.5)
Similarly, the hub weight vector will be the principal right eigenvector of the matrix $M_{AH}$, hence:

$$M_{AH} = (F^{\frac{1}{2}} A)(F^{\frac{1}{2}} A)^T$$  \hspace{1cm} (4.6)

$$h^k = M_{AH} h^{k-1}$$  \hspace{1cm} (4.7)

**Another Example**

(a) Hubs pointing to weak authorities  
(b) Common Authority

![Diagram](image)

**Figure 4.2:** Another problem in HITS.

The solution provided by HubAvg still contains a loophole. In figure 4.2(a) there are two extra red authorities ($A_i$ and $A_j$) that are pointed to by the two hubs ($H_3$ and $H_5$) respectively. HubAvg will assign more weight to the blue authority than the red one, although they are identical. The red hubs are penalized for pointing to the extra two weak authorities. And because they have divided their weights among the authorities that they point to, so, the share of the hubs that point to the weak authorities is less to the red authority in centre, $A_1$. Thus the red authority $A_1$ will be weighted less than the blue authority $A'_1$. This is also an undesirable situation, a node which is simultaneously a strong hub on one topic while a weak hub on another topic, should not be penalized for being a weak hub on another topic.

**Authority threshold**

The effect of such a situation can be lessened simply by applying a threshold operator, that is, to retain only the highest authority weights (depending on threshold). Authority-threshold algorithm sets the hub weight of node $i$ to be the sum of $k$–largest authority weights of the authorities pointed to by $i$. This means a good hub points to at least $k$ good authorities.

$$a_i = \sum_{j \in B(i)} h_j \quad \text{and} \quad h_i = \sum_{j \in F_k(i)} a_j$$  \hspace{1cm} (4.8)

where $F_k(i)$ is a subset of $F(i)$ containing $k$ nodes with highest authority weights. This means that for any node $p \in F(i)$, such that $p \notin F_k(i)$, than $a_p \leq a_q \ \forall q \in F_k(i)$.

**Max operator**

The extreme values of $k$ in authority threshold algorithm provide interesting insight into the algorithm. If $k$ is greater than or equal to the maximum out-degree of the graph $G$ then the authority threshold algorithm
reduces to the original HITS algorithm. And for \( k = 1 \) we get a \( \text{max} \) operator (only the topmost authority weights is set to hub score). \( \text{Max algorithm} \) is to say that a hub node is as good as the \( \text{best} \) authority that it points to \([Tsa04b]\). Applying Max algorithm to the example in figure 4.2(a), the red hubs pointing to the weak authorities would only keep the authority score of the authority node in centre \( A_1 \).

**Norm family**

We want to \( \text{scale} \) the weights, so that lower authority weights contribute less to the hub weight, instead of applying threshold or max operator. Through their weights we can determine the scaling factors. This idea is implemented in the \( \text{Norm family} \) of algorithms. The hub weight of node \( i \) is set to the \( p - \text{norm} \) of the vector of the authority weights of the nodes pointed to by node \( i \) \([Tsa04b; Tsa04a]\). Therefore we have:

\[
 a_i = \sum h_j \quad \text{and} \quad h_i = \left( \sum a_j^p \right)^{1/p} \quad (4.9)
\]

The extreme values of \( p \) in the Norm algorithm also provides interesting insights. For \( p \in [1, \infty] \) as we increase \( p \), the value of the \( p - \text{norm} \) in hub vector is more and more dominated by the highest weights. For example for \( p = 1 \) the norm algorithm reduces to \( \text{HITS} \), and for \( p = \infty \) we will have max operator \([Tsa04b]\). And for \( p = 2 \) the hub vector will be the \( \text{Euclidean norm} \) of authority weights.

The modifications that are applied to only \( \bigcap \) operation of \( \text{HITS} \), can also be applied to its \( \mathbb{I} \) operation, symmetrically. For example, applying Norm operator symmetrically on both hub and authority vectors (instead of just hub vector) will have the same effects on the authority vector as it had on hub vector (in the discussion above). Max operator applied only to hub vector enables the hubs to only receive the top most authority scores of the nodes that they point to. And symmetrically applying max operator on the authority vector enables authority nodes in the graph to receive only the topmost hub scores of the nodes that point to them. Applying max operator on authority vector in figure 4.2(a); the red authority \( A_1 \) in centre will only receive the topmost hub score instead of accumulating all the hub scores.

It requires further analyses to observe the differences between the results of symmetrical application of operators (discussed in last paragraph) and improvements described earlier in equations (4.5), (4.8) and (4.9). Comparing them is out of scope of this study, but we will use the ideas described in this section as a ground to explore improvements in convergences and personalization of the results. In Chapter 5, we would extensively experiment with the algorithms defined in this section together with improvements in the coming sections.

### 4.3 Extrapolation Techniques to accelerate the Convergence

In this section we will probe into a new and genuine formulation of query-dependent LAR using a technique called \( \text{Extrapolation} \). The Extrapolation techniques can be used to accelerate the convergence of Power Method (see Section 3.3.6). The technique is novel as it offers a new approach of taking into consideration important properties of the iterative method for effectively accelerating the computation of the query-dependent family of algorithms (e.g., \( \text{HITS} \), its improvements and \( \text{SALSA} \)).

In mathematics, \( \text{Extrapolation} \) is the process of constructing new data points outside a discrete set of \( \text{known} \) data points. It is similar to the process of \( \text{Interpolation} \), which constructs new points between known points, but its results are often less meaningful, and are subject to greater uncertainty. Interpolation is a specific case of curve fitting, in which the function must go exactly through the data points. In case of convergence, Extrapolation techniques can be employed to accelerate the convergence by using the known data points (values
from successive iterates) to construct new data points (principal eigenvector(s)). Techniques for accelerating the convergent series are often applied in numerical analysis, where they are used to improve the speed of numerical integration, and other well-known series.

Fast convergence and efficient computational speed in query-dependent algorithms are quite crucial, because of the fact that they operate on query time. They should therefore be proficient in terms of time they consume to converge and computations per iteration. The response time in query-dependent algorithms have a direct impact on the overall interaction between user and IR system. For example for an adjacency matrix of size $7000 \times 7000$, it is fairly expensive to compute the operation $\vec{x}_k = A\vec{x}_{k-1}$ several times as $k \to \infty$.

The standard power method is used to compute the principal eigenvector and eigenvalue by computing the successive iterates until convergence. In the power method implementation of the query-dependent algorithms (see Section 3.4) the computations per iteration are usually expensive, depending on the size of matrix. There is a dire need for reducing the number of iterations as there is also an essential need to improve the computations per iteration too.

Many algorithms have been proposed over the years to improve the computational abilities of the power method (for example, Power method with Rayleigh quotient, shifted power method, inverse power method, etc) [Lay94; Gol96]. The fast eigenvector solvers and other numerical improvements involving vigorous use of the matrices are unsuitable for LAR problem, because of the size and sparsity of the matrices.

In this section we will primarily focus on investigating efficiencies in terms of the rate of convergence. In the extrapolation techniques, we largely take advantage of the fact that the principal eigenvalue of the Markov matrix is 1 in theory. This information can be used to compute the estimates of the nonprincipal eigenvectors during the iterations. Through the estimates computed during the successive iterates of Power method, we expect to extrapolate the value of the principal eigenvector.

Extrapolations techniques were previously used by Kamvar et al., [Kam03b], specifically tailored to the PageRank problem, but in our study we have employed it in the query-dependent algorithms such as HITS, its improvements and SALSA.

### 4.3.1 Aitken’s $\Delta^2$ Extrapolation

A technique called Aitken’s $\Delta^2$ (three-points) extrapolation can be used to speed up the convergence of any sequence that is linearly convergent\footnote{A sequence $\{x_i\}$ is said to converge linearly to $x^*$ if there is constant $1 > c > 0$ such that $||x_{i+1} - x^*|| \leq c||x_i - x^*||$ or alternatively $||x_{i+1} - x_i|| \leq c||x_i - x_{i-1}||$} [Cio88]. Aitken Extrapolation method makes use of another popular method called fixed point iteration. If we have an equation of the form $x = f(x)$ we can attain the expected solution simply by iterating through the sequence, $x(k+1) = f(x(k))$. Or more specifically for a given function $f$ defined on real numbers and a given initial point $x_0$ in the domain of $f$, the fixed point iteration is:

\[
x_{k+1} = f(x_k), k = 0, 1, 2, \ldots
\] (4.10)

The series $x_0, x_1, \ldots$ are expected to converge to $x$. If the function $f$ is continuous, then $x$ is a fixed point of $f$, i.e., $x = f(x)$.

The equation (4.10) is the standard fixed point iteration. Now consider the standard power iteration method, we will get a correspondence with fixed point iteration, i.e.,:

\[
\vec{x}^{(k)} = A\vec{x}^{(k-1)}
\] (4.11)
Let us consider $f$ in equation (4.10) as an iterative numerical process, then the intermediate iterates of the linear convergent series, $x_i$, $x_{i+1}$ and $x_{i+2}$ can be used to extrapolate the fixed point $x$. This three-point extrapolation scheme is well known as Aitken $\Delta^2$ extrapolation.

Aitken $\Delta^2$ extrapolation is oldest and most popular extrapolation technique. It forms the basis for other extrapolation techniques. It has also been used to speedup the convergence of the Power method for faster computation of PageRank [Kam03b].

In power method, Aitken acceleration computes the principal eigenvector of the Markov matrix in one step, under the assumption that the power iteration estimate $\vec{x}^{(k-2)}$ can be expressed as the linear combination of the first two eigenvectors, $\vec{u}_1$ and $\vec{u}_2$.

$$\vec{x}^{(k-2)} = \vec{u}_1 + \alpha_2 \vec{u}_2$$

(4.12)

where $\vec{u}_1$ is the principal eigenvector and $\vec{u}_2$ is the second eigenvector of Markov matrix in power method.

The equation (4.12) shows that from the nonprincipal eigenvectors (the values of $\vec{x}^{(k-2)}$ from successive iterates), we can extrapolate the value of the principal eigenvector $\vec{u}_1$. The previous values calculated in the successive iterates could be used to extrapolate the new value (the new data point outside the known data points), the principal eigenvector. This way we could accelerate the rate convergence of the already convergent series produced by Power Method.

The Aitken extrapolation step when applied periodically, enables us to subtract off the estimates of the nonprincipal eigenvectors from the current iterates $\vec{x}^{(k)}$. For the derivation of Aitken acceleration and the empirical proof that it can extrapolate the principal eigenvector for power method see [Kam03b; Cio88]. Aitken extrapolation technique is crucial primarily because the subsequent extrapolation techniques build upon the ideas advocated in this technique. It serves to provide a general premise for extrapolation. It is therefore essential to have a sound appreciation of this technique to comprehend the newer more sophisticated techniques of extrapolation used for accelerating convergence.

In a nutshell we want to use the priori knowledge (which we acquire from the prior iterates) and use that knowledge as a basis to extrapolate the new and better value (the principal eigenvector). We use the assumption that the new iterate(s) can be expressed as a linear combination of the last few iterates. With some changes to this basic assumption various extrapolation techniques can be formulated (for example, Quadratic Extrapolation assumes that last three iterates $\vec{x}^{(k-3)}$, $\vec{x}^{(k-2)}$ and $\vec{x}^{(k-1)}$ together with current iterate $\vec{x}^{(k)}$ can be used to express the new and improved iterate value, see next section). In case of Aitken Extrapolation we are using three-points $\vec{x}^{(k-2)}$, $\vec{x}^{(k-1)}$ and $\vec{x}^{(k)}$ to extrapolate the next point $\vec{u}_1$.

The extrapolation methods are different from standard fast eigensolvers, which mostly relies on the matrix factorization and/or matrix inversion. The extrapolation methods that we study here rely upon the fact that the principal (first) eigenvalue of the Markov matrix is, $\lambda_1 = 1$, in order to find an approximation to the principal eigenvector.

What is crucial here is to identify theoretically and empirically that the extrapolation methods accelerate the convergence, and the computed value is actually the principal eigenvector. In the section below we will provide theoretical proof for the importance of Quadratic extrapolation and its capability to extrapolate the principal eigenvector of the Markov matrix. In Chapter 5 we will provide experimental evidences of the capabilities of Extrapolation to improve the convergent sequence in query-dependent LAR algorithms.
4.3.2 Quadratic Extrapolation

In this section we will closely examine Quadratic Extrapolation technique [Kam03b]. Like Aitken $\Delta^2$ Extrapolation, Quadratic extrapolation also uses the idea of taking the linear combination of last few iterates. Unlike Aitken, Quadratic extrapolation uses the first three (instead of first two) eigenvectors of Markov matrix to express the iterate ($\vec{x}^{(k-3)}$). Therefore, it assumes that the iterate $\vec{x}^{(k-3)}$ can be expressed as the linear combination of first three eigenvectors ($\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$) of Markov matrix.

These assumptions in Quadratic extrapolation enable us to approximate the principal eigenvector in closed form using iterates $\vec{x}^{(k-3)}$, $\vec{x}^{(k-2)}$, $\vec{x}^{(k-1)}$ and $\vec{x}^{(k)}$.

Formulation

From the theory of Power method, we know that the seed vector $\vec{x}^{(0)}$ can be expressed as the linear combination of all the eigenvectors of the Markov matrix (see Section 3.3.6). Thus,

$$\vec{x}^{(0)} = \vec{u}_1 + \alpha_2 \vec{u}_1 + ... + \alpha_m \vec{u}_m$$

While in the specific settings of Quadratic Extrapolation it is assumed that the Markov matrix $A$ in equation (4.11) has only 3 eigenvectors. Based on this assumption the iterate $\vec{x}^{(k-3)}$ can be expressed as linear combination of these 3 eigenvectors.

The quadratic extrapolation can be formulated from this premise that the matrix $A$ has only 3 eigenvectors and therefore we can approximate the iterate $\vec{x}^{(k-3)}$ as:

$$\vec{x}^{(k-3)} = \vec{u}_1 + \alpha_2 \vec{u}_2 + \alpha_3 \vec{u}_3$$ (4.13)

The assumption is not in contrast to reality rather it is used to form the much stronger relation later in the derivation. Of course, the matrix $A$ can have more than 3 eigenvectors. Later in the section we will form a relation based on this assertion. In the experimental analyses in Chapter 5 we have also provided empirical results verifying the validity of this assumption. It is shown in the experiments that Quadratic extrapolation derived from this assumption provides much better rate of convergence than the original algorithms (see Appendix A).

Now we are in a position to derive the required model using equation (4.13). From the assumption (equation (4.13)), the characteristic polynomial $p_A(\lambda)$ of the Markov matrix $A$ can now be written as:

$$p_A(\lambda) = \gamma_0 + \gamma_1 \lambda + \gamma_2 \lambda^2 + \gamma_3 \lambda^3$$ (4.14)

Since the Markov matrix $A$ is stochastic, we know from the theory of Markov chain that the first eigenvalue of $A$ is $\lambda_1 = 1$ [Gol96; Lay94]. Thus:

$$p_A(\lambda = 1) = 0 \Rightarrow \gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 = 0$$ (4.15)

According to Cayley-Hamilton theorem [Gol96] any matrix $A$ satisfies it’s own characteristic polynomial, i.e., $p_A(A) = 0$. Therefore multiplying $\vec{x}^{(k-3)}$ with the characteristic polynomial, we have:

$$p_A(A)\vec{x}^{(k-3)} = [\gamma_0 \mathbf{I} + \gamma_1 A + \gamma_2 A^2 + \gamma_3 A^3]\vec{x}^{(k-3)} = 0$$ (4.16)

---

2 An equation or system of equations is said to have a closed-form solution if, and only if, at least one solution can be expressed analytically in terms of a bounded number of certain “well-known” functions [wik].
This can be simplified as:
\[ \gamma_0 \vec{x}^{(k-3)} + \gamma_1 \vec{x}^{(k-2)} + \gamma_2 \vec{x}^{(k-1)} + \gamma_3 \vec{x}^{(k)} = 0 \] (4.17)

Since we knew from the power iterations:
\[ \vec{x}^{(k-2)} = A \vec{x}^{(k-3)} \cdots \vec{x}^{(k)} = A \vec{x}^{(k-1)} \]

From the above equations and equation (4.17), after simple steps we have:
\[ \vec{x}^{(k-3)} (\gamma_1 - \gamma_2 - \gamma_3) + \gamma_1 \vec{x}^{(k-2)} + \gamma_2 \vec{x}^{(k-1)} + \gamma_3 \vec{x}^{(k)} = 0 \] (4.18)

This can be further simplified as:
\[ (\vec{x}^{(k-2)} - \vec{x}^{(k-3)}) \gamma_1 + (\vec{x}^{(k-1)} - \vec{x}^{(k-3)}) \gamma_2 + (\vec{x}^{(k)} - \vec{x}^{(k-3)}) \gamma_3 = 0 \] (4.19)

Define the following:
\[ \vec{y}^{(k-2)} = \vec{x}^{(k-2)} - \vec{x}^{(k-3)} \] (4.20)
\[ \vec{y}^{(k-1)} = \vec{x}^{(k-1)} - \vec{x}^{(k-3)} \] (4.21)
\[ \vec{y}^{(k)} = \vec{x}^{(k)} - \vec{x}^{(k-3)} \] (4.22)

Inserting equation (4.22) in equation (4.19) gives:
\[ \vec{y}^{(k-2)} \gamma_1 + \vec{y}^{(k-1)} \gamma_2 + \vec{y}^{(k)} \gamma_3 = 0 \] (4.23)

and
\[ \begin{pmatrix} \vec{y}^{(k-2)} \\ \vec{y}^{(k-1)} \\ \vec{y}^{(k)} \end{pmatrix} \vec{\gamma} = 0 \] (4.24)

For the solution of the above system we don’t want to have the trivial solution \( \gamma = 0 \), thus we constrain the leading term of the characteristic polynomial \( \gamma_3 \) as:
\[ \gamma_3 = 1 \] (4.25)

After substituting the value of \( \gamma_3 \), equation (4.24) can be written as:
\[ \begin{pmatrix} \vec{y}^{(k-2)} \\ \vec{y}^{(k-1)} \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -\vec{y}^{(k)} \] (4.26)

Hence we have an over determined system of linear equations:
\[ \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -\mathbf{Y}^\dagger \vec{y}^{(k)} \] (4.27)

Here \( \mathbf{Y}^\dagger \) is the pseudoinverse of the matrix shown in the left side of the equation (4.26), \((\vec{y}^{(k-2)}, \vec{y}^{(k-1)})\).

From the equation (4.14) and above equations we can therefore find the coefficient of the characteristic polynomial \( q_A(\lambda) \).

We may divide the characteristic polynomial with \( (\lambda - 1) \) to get \( q_A(\lambda) = p_A(\lambda)/(\lambda - 1) \). Hence we have now:
\[ q_A(\lambda) = (\gamma_0 + \gamma_1 \lambda + \gamma_2 \lambda^2 + \gamma_3 \lambda^3)/(\lambda - 1) = \beta_0 + \beta_1 \lambda + \beta_2 \lambda^2 \] (4.28)
By polynomial division and after some simple algebraic operations we get the values for beta:

\[
\begin{align*}
\beta_0 &= \gamma_1 + \gamma_2 + \gamma_3 \\
\beta_1 &= \gamma_2 + \gamma_3 \\
\beta_2 &= \gamma_3
\end{align*}
\]  
(4.29) (4.30) (4.31)

By Cayley-Hamilton theorem, for any vector \( \vec{z} \) in \( \mathbb{R}^n \) we also have:

\[
q_A(A) \vec{z} = \vec{u}_1
\]  
(4.32)

where \( \vec{u}_1 \) is the principal eigenvector of matrix \( A \) corresponding to eigenvalue \( \lambda_1 = 1 \). Thus by letting \( \vec{z} = \vec{x}(k-2) \):

\[
\vec{u}_1 = q_A(A)\vec{x}(k-2)
\]  
(4.33)  
\[
= [\beta_0 + \beta_1 A + \beta_2 A^2] \vec{x}(k-2)
\]  
(4.34)  
\[
= \beta_0 \vec{x}(k-2) + \beta_1 A \vec{x}(k-2) + \beta_2 A^2 \vec{x}(k-2)
\]  
(4.35)

Using the power iterations in above equations:

\[
\vec{x}(k-2) = A \vec{x}(k-3) \ldots \vec{x}(k) = A \vec{x}(k-1)
\]

Thus we get the closed form solution for \( \vec{u}_1 \), as:

\[
\vec{u}_1 = \beta_0 \vec{x}(k-2) + \beta_1 \vec{x}(k-1) + \beta_2 \vec{x}(k)
\]  
(4.36)

The equation (4.36) together with equations (4.29) - (4.31) and equation (4.27) can be used to implement the Quadratic extrapolation. Hence together they will help to provide an approximation to the principal eigenvector of the Markov matrix \( A \). The above derivation steps are inspired from the work in [Kam03b; Cio88].

**Discussion**

In the above formulation in equation (4.27) we have to solve an overdetermined system of linear equations:

\[
\begin{pmatrix}
\gamma_1 \\
\gamma_2
\end{pmatrix} = -Y^T \vec{y}^{(k)}
\]

The above overdetermined system can be solved through any least-square method, for example, through the QR factorization of Gram-Schmidt algorithm [Gol96; Lay94]. The Quadratic extrapolation technique can be further optimized by applying a better solver to the overdetermined system above.

The Quadratic Extrapolation improves convergence much better than the original rate of convergence of the algorithm, based on the empirical results (see Chapter 5 and Appendix A). In the slow convergent series, the Quadratic Extrapolation is proved to be an effective technique. For example, if the second eigenvalue of the Markov matrix is close to 1, i.e., \( \lambda_2 \to 1 \), theoretically and empirically the convergence of power method tends to slow down. In such a situation the slow converging sequence of Power method can be accelerated radically by Quadratic Extrapolation (see Section 5.2.1).

The important thing in Quadratic Extrapolation is that it should be applied periodically. Once the parameters for Extrapolation (such as \( \vec{x}(k-3) \), \( \vec{x}(k-2) \), \( \vec{x}(k-1) \) and \( \vec{x}(k) \)) are ready we could either apply Quadratic extrapolation step immediately or apply it at any other instance with appropriate values. It doesn’t necessarily
need to be applied too often to achieve maximum benefit. Experiments in Chapter 5 reveal interesting insights about the potential of Quadratic Extrapolation in various settings. By manipulating the periodic application of Quadratic extrapolation we can administer the convergence behaviour of an algorithm.

Theoretically, Quadratic extrapolation technique is used to subtract off the errors in the current iterate along the direction of the second and third eigenvectors, as mathematically represented by equation (4.13). By doing that it enhances the convergence for the future application of the power method. The approximate principal eigenvector as a result of Extrapolation step serves as a good approximation for the further iterates, which help to converge much faster. For more detailed discussion on the experimental manifestation of the extrapolation techniques see Section 5.2.1.

In the next section we will explore another interesting technique for extrapolation, where some important properties of the Markov matrix are exploited to make a more generic and cleaner formulation of Extrapolation.

### 4.3.3 Power \((A^d)\) Extrapolation

Based on the ideas initially put forward in Aitken Extrapolation and Quadratic Extrapolation discussed in previous sections Haveliwala et al., [Havb] construct another interesting formulation of extrapolation. Similar to both Aitken and Quadratic extrapolation, here also by subtracting off the errors along several nonprincipal eigenvectors from the current iterates, it is intended to accelerate the rate of convergence of Power method. Not just relying on the values of successive iterates but other important properties of Markov matrix, i.e., the nonprincipal eigenvalues could be exploited to accelerate the convergence.

In linear algebra, finding the nonprincipal eigenvalues of a Markov matrix is a problem in itself. Calculating nonprincipal eigenvalues of Markov matrix may increase the computational overheads, instead of providing any improvements. Thus apparently the idea of using the nonprincipal eigenvalues for acceleration may not seem conducive in general. But in a study by Haveliwala and Kamvar [Hava], they discovered interesting insights about the nonprincipal eigenvalues of the Markov matrix in PageRank algorithm. They have proved that the modulus of second eigenvalue of the Markov matrix (or the Google matrix) is given by the damping factor ‘\(c\)’ in PageRank algorithm (see Section 3.3, where \(\alpha = c\), in the original formulation). Thus if the row stochastic matrix \(S\) in equation (3.6) has at least two irreducible closed subsets, then the second eigenvalue of \(S\) is given by:

\[
\lambda_2 = c
\]

Note that the webgraph can have many eigenvalues with modulus of ‘\(c\)’ (i.e., one of \(c, -c, ci, \) and \(-ci\)). These eigenvalues of the webgraph has been exploited in the power extrapolation to approximate the principal eigenvector of the hyperlink matrix \(A\) corresponding to the webgraph. In the next section we will briefly formulate the use of second eigenvalue for accelerating the convergent series of power method, and in Chapter 5 we will also present experimental findings for this approach.

**Formulation**

In Power Extrapolation the iterate \(\tilde{x}^{(k-2)}\) can be represented as linear combination of three eigenvectors \((\tilde{u}_1, \tilde{u}_2, \text{ and } \tilde{u}_3)\) of Markov matrix. Making use of the same assumption as it was in Quadratic Extrapolation. Thus:

\[
\tilde{x}^{(k-2)} = \tilde{u}_1 + \alpha_2 \tilde{u}_2 + \alpha_3 \tilde{u}_3
\]

(4.37)
The nonprincipal eigenvalues corresponding to nonprincipal eigenvectors $\vec{u}_2$ and $\vec{u}_3$ are ‘c’ and ‘−c’ respectively, according to the results in [Hava]. From Power Iterations we have:

$$\vec{x}^{(k)} = A^2 \vec{x}^{(k-2)} \quad \text{(since} \quad \vec{x}^{(k)} = A (A \vec{x}^{(k-2)}) \quad \text{as} \quad \vec{x}^{(k-1)} = A \vec{x}^{(k-2)})$$

Substituting the above relations in equation (4.37) we get:

$$\vec{x}^{(k)} = A^2 (u_1 + \alpha_2 u_2 + \alpha_3 u_3)$$

(4.38)

$$\vec{x}^{(k)} = u_1 + \alpha_2 \lambda_2^2 u_2 + \alpha_3 \lambda_3^2 u_3$$

(4.39)

We can replace $\lambda_2 = c$ and $\lambda_3 = −c$ in equation (4.39), thus we have:

$$\vec{x}^{(k)} = u_1 + c^2 (\alpha_2 u_2 + \alpha_3 u_3)$$

(4.40)

$$\vec{x}^{(k)} = u_1 + c^2 (\vec{x}^{(k-2)} - u_1)$$

(4.41)

From equation (4.41) the closed form of the approximated principal eigenvector $\vec{u}_1$ will be:

$$\vec{u}_1 = \frac{\vec{x}^{(k)} - c^2 \vec{x}^{(k-2)}}{1 - c^2}$$

(4.42)

The above derivation leads to $A^2$ Extrapolation which subtract off error along the eigenspaces corresponding to eigenvalues $c$ and $−c$. There is a general derivation as well to the case where the eigenvalues of modulus of $c$ given by $c \times d$, where $d$ are the $d^{th}$ root of unity, are used to form a generalized closed form based on variable $d$ [Havb]. For example, for $d = 4$ the nonprincipal eigenvalues of modulus of $c$ are given by $c, ci, and −ci$, which means $4^{th}$ roots of unity.

The generalized case will have the closed form of the principal eigenvector $\vec{u}_1$ as:

$$\vec{u}_1 = \frac{\vec{x}^{(k)} - c^d \vec{x}^{(k-d)}}{1 - c^d}$$

(4.43)

For details about the derivation of equation (4.43) see [Havb].

The implementation of $A^d$ Extrapolation is much more simpler than the Quadratic Extrapolation, just to implement equation (4.43). Theoretically the overhead due to $A^d$ extrapolation is negligible since it is applied only once. The convergence is also found to be similar to the Quadratic Extrapolation, but the wallclock-speedup is higher in $A^d$ extrapolation [Havb]. In contrast to the findings by Haveliwala et al., our results show that the convergence behaviour due to power extrapolation in query-dependent algorithms are not comparable to that of Quadratic extrapolation in the empirical settings, see Chapter 5

**Discussion**

In the case of PageRank the *Eigengap* $(1 - |\lambda_2|)$ for the Markov matrix $A$ is given exactly by the teleportation probability $1 - c$, in accordance to the findings by Haveliwala et al., discussed above. Theoretically, if the second eigenvalue $\lambda_2$ is close to 1, then the convergence of the power method will be slow. Because of the fact that convergence of the power method depends on $|\lambda_2|/|\lambda_1|$ factor, and $k$ must be fairly large before $((|\lambda_2|/|\lambda_1|)^k$ converge to 0 [Lay94; Gol96]. In PageRank reducing the factor $|\lambda_2|/|\lambda_1|$ correspond to the reduction of the damping factor $c$, because we can only change the numerator which is $|\lambda_2|$. But reduction of damping factor means increase in teleportation $(1 - c)$ and hence increasing the chance for the spammers to inflict the rankings. However a high teleportation probability ($(|\lambda_2|/|\lambda_1|) → 1$) constitutes slow convergence of power method.
Under such circumstances it is highly rewarding to accelerate the slow convergence of power method. The methods of Extrapolation discussed above supposedly provide a faster convergence even with high teleportation probabilities.

The $A^d$ Extrapolation is based on second eigenvalue of the Markov matrix. In HITS and other query-dependent algorithms such as SALSA, the power extrapolation cannot be directly applicable. The second eigenvalue of the Markov matrix should be calculated independently. It is not a trivial problem to compute the nonprincipal eigenvalues of Markov matrix in general case. The teleportation scheme cannot be directly employed in the query-dependent algorithms, unless we incorporate the damping factor the same way as we did in PageRank (see Section 3.3.5).

There is a phenomenon called Deflation in linear algebra which can be used to compute the nonprincipal eigenvalues of the Markov matrix. Deflation is a technique of reducing the dimension of Markov matrix corresponding to its dominant eigenvalue. Once we compute the dominant eigenvalue, the Markov matrix can be reduced to one lower dimension (by subtracting the column corresponding to the dominant eigenvalue). We can now compute the dominant eigenvalue of the reduced dimension matrix, which will be the second eigenvalue of the original Markov matrix. Repeat this process until all the eigenvalues are computed. Finding the nonprincipal eigenvalues of Markov matrix is therefore not an easy task. The main techniques for dimensionality reduction are principal component analysis (PCA) and the famous singular value decomposition [Lay94].

In the Section 4.4 we will discuss one way to incorporate the damping factor into the query-dependent HITS and possibly SALSA too. After that modification it will be then possible to apply power extrapolation under the same assumption as we had applied for PageRank algorithm.

### 4.3.4 Insights into Extrapolation

Extrapolation is one of the effective techniques in numerical analysis, but its use for acceleration of convergence in power method is novel. There are a lot of things to explore in the topic of Extrapolation. The studies so far just provide a definition level insight into the topic. There could be a lot of different and useful insights into much dynamic aspects of Extrapolation.

#### A new premise

The Extrapolation methods described in the previous sections are built upon the premise constructed about the initial function (equations (4.12), (4.13) and (4.37)). The common thread among all the extrapolations techniques is the initial assumption. In case of Quadratic Extrapolation we assumed that the matrix $A$ has 3 eigenvectors, and expressed the current iterate as the linear combination of these 3 eigenvectors. A quadratic function illustrates a much closer representation to the reality in this case, and hence provides much better convergence.

One of the possible prospects in Extrapolation could be to start with another new premise. A much deeper understanding of Markov matrix and the insight into properties of LAR algorithms in question could be valuable to form a new premise(s). It could be possible to also consider the personalization (next section) factor in the construction of the new premise in Extrapolation. A possible future work from this study could be to explore extrapolation independently with the focus on formation of a new premise.
Extrapolation parameters

From experiments in the Chapter 5 we found that from the behaviour of the convergence graph we could manipulate the extrapolations parameters to control the convergence. In this regard the number of times we apply extrapolation step is quite crucial. If extrapolation is applied more often than required it might increase the overhead instead of improving. We will discuss in detail about the exploitations of the parameters in extrapolation during experimental analyses in the next chapter.

Hybrid Extrapolation technique

We might as well think of extrapolation in a ‘hybrid’ environment. That is, considering the properties of different extrapolation techniques and depending on the behaviour of the graph we could apply different extrapolation techniques at different instances during iterations of the power method. We will experiment with the hybrid of Quadratic Extrapolation and Power Extrapolation to experimentally examine this approach. But it also requires an independent study to come up with a possible framework which could be used to exploit different extrapolation techniques in a hybrid environment in order to achieve a much controlled convergence in power method. And to observe more closely the dependences and dynamics of different extrapolation techniques applied simultaneously. It could turn out to be a novel approach towards active use of extrapolation. And again, the factor of personalization can also be employed in the hybrid extrapolation scheme.

The findings discussed in this section just depict limited implications of extrapolation. There are quite a lot of other possibilities too for further innovation in the field of Extrapolation.

4.4 Personalization

There is no exact mathematically precise measure of “best”, indeed it lies in the eyes of the beholder.

Personalization is one of the very significant and key topics in the relevancy ranking. A user is conventionally expected to solicit queries like: “what sources are popular or authoritative on some topic (query)?”. In contrast, the Web user, with sophisticated or individualistic needs might query: “what sources are authoritative, given that some sources are preferred, which might not be authoritative in general”. A user therefore may have opinions or inclinations on certain topics and may consider them better than the authorities in general on the topics in question. Under such a condition a user would like to see authoritative sources that coincide with their own preconceived conception of authoritativeness for certain topics. Or in general a mechanism is expected through which users can enforce their opinions on the search outcomes. Personalization is therefore the mechanism to bridge the user’s opinions and conceptions (preferences in general) with the system’s measure of authoritativeness.

The user of IR system now doesn’t want to see the search outcomes the way system produces it, rather they need personalized results, specific to their own behaviours and moods. The user expects that the IR systems somehow implicitly monitor their behaviours, organize and update their behavioural information (the implicit feedbacks), and when they query, the search outcomes should be in accordance to their behaviours. For example, for the search query “movies”, the page ‘www.imdb.com’ (Internet Movie Data Base) might be considered as top authority by most of relevancy ranking algorithms, but a user interested in theatres might want ‘www.amctheatres.com’ (AMC Theatres) as the most relevant hit to his/her query “movies”.

Users on the Web usually have a set of preferred pages (e.g., bookmarks, usage history or click-through) and
they require that the search engines capture these information and based on that prepare the search outcomes, instead of producing generally relevant pages.

The *one-size fit all* approach by the conventional search-engines requires a placeholder or accommodation for personalization. Hence a search by user A should have different outcomes than a search by user B. The users want to *inject* their own opinions in the search process, about which type of sources are considered important by them, and what should the search outcomes look like. The IR model must therefore cater the user needs more advantageously by either implicitly decipher the trends in their searching behaviours or explicitly utilize any other relevant information, such explicit user relevance feedback.

The personalization is not only centred towards users’ needs and preferences, but also through personalization the IR system can be in a better control of the search outcomes. One of the ways of making money by the search engines today is to boost certain sources based on some query topics, e.g., for advertising purpose. Advertising accounted for over 97% of Google revenue in 2003 [Lan06]. But also personalization is used to counter the bigger problem of *spamming* to a certain extent. Through personalization the possible spamming sources can be prevented to inflict the rankings, by assigning them negative weights with the facility of personalization schemes. Thus personalization can better serve the cause of *anti-spamming* and could be extensively and strategically employed to create firm barriers for spammers.

There has been numerous recent developments done in personalizing PageRank algorithm using several approaches, such as [Jeh03; Hav03b; Hav03a; Kam03a; Ric02]. The initial idea of personalizing PageRank was first proposed by Page et al., [Pag98], but was never fully explored. There has been some marginal amount of work done to personalize HITS too, such as, the *Gradient Ascent* approach by Chang et al. [Cha00].

The user search or usage history (e.g., usage logs) can be mapped to the set of representative content categories drawn from some sources, e.g., *Open Directory Project*\(^3\). The topic-sensitive approach by Haveliwala [Hav03a], take the linear combination of 16 topic-sensitive PageRank vectors (16 topics taken from Open Directory Project) to calculate the overall *Personalized PageRank vector*.

“To encompass different notions of importance for different users and queries, the basic PageRank algorithm can be modified to create ‘personalized views’ of the web, redefining importance according to user preference” [Jeh03].

These studies are just a small subset of the ongoing research in personalization. In this section we will go through a few of the important approaches towards personalization. We will then present a general approach towards personalization which can be tailored to bring the query-dependent models like HITS and SALSA within the framework of personalization. At the end we will also present some of the approaches of personalization in HITS algorithm, and some other approaches which offer the possibilities for personalization in HITS and its descendants.

**Personalization in PageRank**

In the Section 3.3.5 we derived equation (3.7) after the necessary adjustments, the factor \( \frac{1}{n} ee^T \) in that equation can be called as the *teleportation matrix*, \( E \). Thus we could rewrite equation (3.7) as:

\[
G = \alpha S + (1 - \alpha)E
\]

The adjustments to the basic PageRank model hence created a possibility for personalization with the help of the teleportation matrix \( E \). Initially \( E = \frac{1}{n} ee^T \), but in the modified (personalized) version, \( E \) will be an

\(^3\)Web directory for over 2.5 million URLs, http://www.dmoz.org/
$n \times n$ rank-one row stochastic matrix:

$$E = e\vec{v}^T$$

where $\vec{v}$ is an $n$-vector whose elements are all non-zeros containing probability distributions, hence it is called the *personalization vector*. In terms of personalization vector $\vec{v}$, the PageRank model will be the same as in the Section 3.3.5, except the last step of derivation, which could be replace by:

$$G = \alpha A + (\alpha a + (1 - \alpha)e)\vec{v}^T$$  \hspace{1cm} (4.45)

Equation (4.45) signifies that; at each time step, a random surfer who hops from webpage to webpage, with probability $\alpha$, he randomly chooses the next page from the set of out-links of the current page. Occasionally with a probability $(1 - \alpha)$ the surfer gets bored (or tend to avoid dead-ends or endless cycles), instead of following out-links, he jumps to a random webpage, i.e., he teleports. The destination of the random jump will be chosen according to the probability distribution given in $\vec{v}$. The preferences of the user therefore can be stored in the probability distribution $\vec{v}$. Thus $\vec{v}$ will automatically give preference to those pages which are of interest to the user.

In terms of the basic PageRank model the probability distribution for the random jump is set to be *uniform*, i.e., when the random surfer gets bored, he jumps to a random page based on uniform distribution (e.g., $\frac{1}{n} \vec{1}$). If that probability distribution is set to be *non-uniform*, the resultant PageRank vector can be biased to prefer certain kinds of pages. That is why probability vector $\vec{v}$ is called as *personalization* vector.

Through *biasness* (in terms of probability distribution) in the probability vector $\vec{v}$, we could personalize the outcomes of the search. This is how the *user-independent* PageRank can be specialized to the *user-dependent* PageRank, with the personalization based on users’ own choice. The only limiting factor here will be the computational cost of calculating the personalized PageRank for every user, which will be fairly prohibitive. We will talk about the computational concerns of personalization in the later sections.

### 4.4.1 Intelligent Surfer

The random surfer in personalized model can also be called as an *intelligent surfer*, one that is guided by a controlled probabilistic model of the relevance of a page to a query. This means that when a random surfer wants to select a destination at the time of jump, he will select them *intelligently*, instead of randomly.

By enriching the user query with their contextual information such as their behavioural trends (preferences) we could enforce the intelligent surfer to focus on the user specific outcomes. An intelligent surfer is expected not to loose the query context (memory of query topic) as quickly as the undirected or unintelligent random surfer. The jumps of the intelligent surfer must be guided by the contextual information in the personalization vector.

In the study by Richardson and Domingos [Ric02], a *query-dependent, content-sensitive* version of PageRank has been introduced. The random surfer (or *directed surfer*) in that case, when choosing among multiple *out-links* from a page, it tends to follow those which lead to pages whose content has been deemed relevant to the query. Similarly, at the time of random jump the directed surfer tends to choose the page which is content-wise more relevant to the query at hand. Thus a query *enriched* with user specific details would automatically personalize the outcomes, by precisely using the most representative content depicting the user’s preferences.

Although the idea initially presented in directed surfer model doesn’t intended to cope with personalization. The concept can be used to personalize the outcomes based on content-sensitive approach. If content-wise the

---

4For example by using the traditional IR measures like $TF – IDF$ see Section 2.3.2 as the content dependent relevance function.
query contains the user specific details, then the directed surfer would automatically take care of personalizing the search outcomes. This could be a possible future work to make the query subsystem in an IR model accountable for the task of enriching the user query with user related information. The work done in directed surfer model by Richardson and Domingos would then be used for constructing the personalized results. In the modified directed surfer model (expected), the task of personalization is shifted from the relevance subsystem to the query subsystem.

### 4.4.2 General Formulation of Personalization

To formulate a general representation for personalization using the random surfer model of PageRank, we need to consider equation (4.44) in the previous section. We will use this formulation as a basis to construct a general model for personalization, which could be subsequently used in the query-dependent algorithms such as HITS and SALSA.

Taking transpose of equation (4.44):

\[ G^T = \alpha A^T + (1 - \alpha)\vec{v}\vec{e}^T \]  

(4.46)

From the power iterations we have:

\[ \vec{x} = G^T \vec{x} \]  

(4.47)

Substituting equation (4.46) in equation (4.47) we get:

\[ \vec{x} = (\alpha A^T + (1 - \alpha)\vec{v}\vec{e}^T)\vec{x} \]  

(4.48)

\[ \vec{x} = \alpha A^T \vec{x} + (1 - \alpha)\vec{v}\vec{e}^T \vec{x} \]  

(4.49)

\[ \vec{x} = \alpha A^T \vec{x} + (1 - \alpha)\vec{v} \quad \text{(since } \vec{e}^T \vec{x} = 1, \text{ as } \vec{x} \text{ is probability vector}) \]  

(4.50)

\[ \vec{x} - \alpha A^T \vec{x} = (1 - \alpha)\vec{v} \]  

(4.51)

\[ (I - \alpha A^T)\vec{x} = (1 - \alpha)\vec{v} \]  

(4.52)

\[ \vec{x} = (1 - \alpha)(I - \alpha A^T)^{-1}\vec{v} \]  

(4.53)

Since the matrix \((I - \alpha A^T)\) is strictly diagonally dominant therefore the inverse exists [Lay94; Gol96]. Because:

\[ (I - \alpha A^T)^T = (I - \alpha A^T) \]

Let

\[ Q = (1 - \alpha)(I - \alpha A^T)^{-1} \]

Substituting \(Q\) above in equation (4.53), we have:

\[ \vec{x} = Q\vec{v} \]  

(4.54)

The vector \(\vec{x}\) is the Personalized Rank Vector corresponding to the personalization vector \(\vec{v}\). In linear algebra, columns of matrix \(Q\) in equation (4.54) provide the complete basis\(^5\) for the personalized rank vectors. Because any personalized rank vector can be expressed as a convex combination (hence, linear combination) of the columns of \(Q\) from equation (4.53).

\(^5\)A basis is a set of vectors that, in a linear combination, can represent every vector in a given vector space, and such that no element of the set can be represented as a linear combination of the others. In other words, a basis is a linearly independent spanning set [Lay94].
In case of PageRank, the calculation of the matrix $Q$ is computationally prohibitive. Taking the inverse of huge webgraph’s adjacency matrix $A$ would be a computational nightmare.

However low-rank approximation of $Q$ can be still be used to achieve personalization partly. In that case we won’t express all the personalized rank vectors, but rather focus on those corresponding to the convex combinations of the rank vectors in the reduced basis set.

Equation (4.54) serve to be the general model for personalization which could be moulded in other algorithms as well, not just PageRank. We will study different approaches that exploit the general model of personalization in the next section. We will also present other motivating models that attempt personalization in other prospects. That is, instead of just employing a personalization vector, the adjacency matrix can be equipped with personalization information.

### 4.4.3 Approaches towards Personalization

**Topic-sensitive approach for Personalization**

Introduced by Haveliwala [Hav03a], topic-sensitive approach computes a set of the PageRank vectors, based on a set of representative topics. This captures more accurately the notion of importance with respect to a particular topic. Thus instead of using single global PageRank vector, he takes linear combination of the pre-calculated topic-sensitive PageRank vectors at query time, weighted using the similarities of the query to the topics.

Topic-sensitive PageRank scheme was devised to compute an $n \times k$ approximation to $Q$ in equation (4.54) using 16 topics. This scheme of personalization therefore uses a very coarse basis set to personalize the rankings. But its not truly personalizing for each and every individual rather the personalization is based on the topic of the query and query context.

The columns of the reduced-rank matrix $Q$ are generated offline independent of any query (the topic-sensitive PageRank vectors). At the query time the convex combination of the columns of reduced-rank matrix $Q$ is taken, using the context of the query to compute the appropriate topic weights.

In terms of the equation (4.54) the vector $v_j^i$ ($j$th column of $Q$) generated by the topic-sensitive scheme is a dense vector, generated using a classifier for topic $T_j$; hence $(v_j)_i$ represents the (normalized) degree of membership of page $i$ to topic $j$. Thus in this scheme a random surfer can teleport to a topic $T_j$ with some probability $w_j$, followed by teleport to a particular page $i$ with probability $(v_j)_i$. That is how the problem of heavily linked pages getting highly ranked for the queries for which they have no authority, can be avoided. For more information see [Hav03a; Hav03b].

**Dynamic programming and Personalization**

In the study by Jeh and Widom [Jeh03], dynamic programming algorithms for computing the partial vectors were employed, in order to construct personalized view from the partial vectors. Partial vectors are encoded form of personalized views. In this case the algorithms compute an $n \times k$ low-rank matrix $Q$ (i.e., each columns corresponding to the partial vectors) corresponding to the highly ranked pages. A suitable number of partial vectors are computed offline and at query time those vectors are “constructed quickly” to form the Personalized PageRank score (personalized rank vector, in equation (4.54)).

This approach could be considered analogous to the topic-sensitive approach; here partial vectors are calculated using the concepts of dynamic programming while in topic sensitive approach personalized vectors are calculated based on the selected topics. Going into minor details of this algorithm is out of scope of this study. For more information about this approach, see [Jeh03].
Graph structure and Personalization

There is another study by Sepandar et al. [Kam03a], they have identified the web as having nested block structure. That is, the vast majority of the hyperlinks on the Web, link pages on a host to other pages on the same host. There are chunks of pages which point mostly to each other in the same host instead of pointing to other hosts. Most of the links on the web are therefore inter-host links rather than intra-host links. The overall web’s network structure has clusters or chunks of pages connected tightly together within each chunk. They have formed a block structure where each block is loosely connected with others and tightly connected within itself.

The graph structure of web has been analyzed by Broder et al., [Bro00] where they also think that the connectivity in web is strongly limited by a high-level global structure (see figure 3.5). The graph structure of web has been characterized by the “bow-tie” structure. It is suggested that exploiting the “bow tie” structure of the web would be fairly useful in computing PageRank. The webgraph is also proved to follow the power law distribution, i.e., “the number of web pages with in-degree $k$ is proportional to $k^{-\beta}$”. For more information on web’s graph structure, see [Lu04; Bro00; Ara02; Cha02].

The study of graph structure of web and its implication on IR is also very exciting and prospective. Discovering, classifying and organizing structures in the webgraph are quite evolutionary and intrinsically innovative. The study of webgraph is peculiar and essential because through such studies we try to measure the distribution of broad topics and topical clusters on the Web. The knowledge of different techniques such as aggregation/disaggregation, composition/decomposition and classification of webgraph would significantly stimulate the new advances. These techniques and other strategies have clearly predicated on our understanding of the social processes which shape the webgraph. Here we are more concerned with the indirect but essentially valuable use of the graph structure of web, i.e., the implication of graph structure of the web on personalization.

The BlockRank algorithm introduced by Sepandar et al. [Kam03a], exploits the block structure of the web by calculating the local PageRank vector of each block. These local PageRank vectors also corresponds to the columns of the low-rank matrix $Q$. The local PageRank vectors are weighted according to their blocks. These weighted local PageRank vectors (columns of $Q$) are aggregated and the resultant vector is used as an starting vector to the standard PageRank algorithm.

The analogy of the random surfer model to the BlockRank algorithm will be; once the random surfer gets bored he will teleport to a particular block or host $B_j$ with probability $w_j$, instead of arbitrary set of pages. And following that he will further teleport to page $i$ in the block $B_j$ with probability $(v_j)(i)$. For example, a surfer interested in sports may teleport from one host page, e.g., ‘www.espn.com’ to another host, e.g., ‘www.cnn.com’ instead of teleporting to the same host. The primary purpose of the BlockRank algorithm is to exploit the block structure of the webgraph in order to achieve speedup in PageRank. Locality of references in each block reduces disk $i/o$ and allows parallel computation of the local PageRank vectors and sometimes reuse of the old values too.

The implication of the graph structure of web for personalization is an innovative approach. The BlockRank method provides a broad perspective of the use of graph structure which can be indirectly used for personalization. But a more active and thorough use of graph structure for personalization can give more insight into the problem of personalization. For example, if the adjacency matrix of the webgraph can possibly contain user specific personalization data, that kind of data could be used to generalize and personalize the user preferences. Hence personalization based on graph structure either fetches the preferred pages or the pages related to the preferences of users.
A further in depth study of purely graph structure of web together with personalization can reveal interesting features and characteristics of graph structure in exposition with personalization and additional magnificence in personalization.

### 4.4.4 Personalized and Stabilized HITS

The basic HITS algorithm doesn’t support user-specific personalization; therefore it is worth considering personalization in HITS and in other query-dependent alternatives such as SALSA. HITS does provide authoritative results on query topics, but it does not support personalized outcomes according to users specific preferences. It does measure a general notion of authority but does not cater the user’s internal notion of authority of which sources are important.

One of the ways for personalization in HITS would be to restructure it according to the general model for personalization given in Section 4.4.2. In order to do that we have to incorporate the random surfer model into HITS, i.e., random surfer(s) follows the hyperlinks and occasionally, depending on the personalization vector, jump to another destination (based on non-uniform distribution). In other words we have to have adjustments in HITS analogous to the ones in the case of PageRank, see Section 3.3.5.

We will explore some techniques through which we could possibly incorporate user preferences in the model and hoist the authoritativeness of the sources considered favourable by user. We will focus on implicit and explicit properties of HITS and also some of its limitations in order to exploit them and formulate a personalization scheme in the algorithm. The focus is therefore to bring the query-dependent HITS and SALSA within the framework of personalization.

#### Gradient Ascent

There could be two possible ways to cater personalization in general; one is to enrich the user query with his personalization information, and second is to alter the weighting of the link matrix to influence relevance of the retrieved documents, as discussed in previous sections. In this subsection and in the next we will focus on the second approach, i.e., to manipulate the entries in the link matrix and use them for personalization.

Here we would like to have a mean of augmenting the information in the link matrix $A$ with the users’ preferences to personalize the retrieved pages. This way we could boost the weights of sources which are considered important by the user. By doing that it is expected that the system generalizes the users’ preferences by not just boosting the weights of sources in question, but also boost the scores of the related sources as appropriate. It is true in HITS that if we alter the link matrix (by incorporating user preference), the mutual reinforcement relation will automatically distribute the authorities among the neighbourhoods of the boosted pages.

In the study [Cha00], Chang, Cohn and McCallum believe that instead of relying on automatically distribution of authority by HITS, we can perform gradient ascent on the elements of the link matrix. The automatic distribution of authority by HITS suffers from the intrinsic prejudice exist in the hyperlink structures. The TKC effect of HITS will possibly forbid a fair distribution of authorities. Instead of relying on distribution based on tightly linked pages, the geometry of hyperlink space (eigenvector space) is considered for taking care of spread of authorities.

---

6 Gradient ascent is an optimization algorithm. To find the local maximum of a function using gradient ascent, one takes steps proportional to the gradient of the function at the current point.
This way we could possibly alter the link matrix in order to more closely align its principal eigenvector with the preferences of the user (recall the theory of linear trend in HITS, see figure 3.7). To perform gradient ascent step to the link matrix, we simply add a fraction \( \gamma \) of the gradient \( \forall k,i \Delta A_{k,i} = A_{k,j}a_i \) to boost document \( j \) to each element of the link matrix, hence:

\[
A_{k,i}^+ = A_{k,i} + \gamma \cdot \frac{\Delta A_{k,i}}{\sum_i \Delta A_{k,i}}
\] (4.55)

The idea of spread of authorities in gradient ascent approach is the same as the spread of activation model defined earlier in 1984 by Anderson and Pirolli [AND84]. Which is also later used in individual studies [Pir96; Car97], in order to create a better visualization of information on web. For detailed information on gradient ascent approach see [Cha00; Tan02].

In the next section we will use the same approach as in this section – we will alter the link matrix, and then apply the personalization scheme.

**Exponentiated Input**

As identified in [Mil01; Far06; Ng01b], the HITS algorithm doesn’t always behave as expected. It is possible that HITS can return non-unique weight vector and can inappropriately assign zero weights to certain sources in the network (see experimental results in Section 5.2.2).

Technically, the dominant eigenvalue of the authority matrix \( A^T A \) can be repeated, which causes the authority vector to be non-unique. It is not always the case to get a unique dominant eigenvalue for the matrix \( A^T A \). In terms of linear algebra this means that the authority matrix \( A^T A \) is reducible [Lay94]. That is, there are set of states that it’s possible to enter, but once entered it’s impossible to exit.

When the principal eigenvalue is repeated, the ranking vector or principal eigenvector can be any non-negative vector in multi-dimensional dominant eigenspace. Repeated principal eigenvalue will have the corresponding multiple principal eigenvectors, which will span the dominant eigenspace. For example, for a two-level reversed binary tree whose edges point upward towards the root, the eigenvalue of the authority matrix \( A^T A \) can be repeated, see the examples in Section 5.2.2. There are some characteristic graphs that give rise to the repeated eigenvalues problem. We have experimented with some of them in Chapter 5, also see [Far06; Mil01].

The second problem is that HITS algorithm yields zero authority weight for apparently important nodes in certain types of graph see Section 5.2.2. If the dominant eigenvalue of matrix \( A^T A \) is repeated, or the principal eigenvector has zero entries for nodes with non-zero out-degree, then the matrix \( A^T A \) must be reducible (proof available in [Far06]). It has been formally proved in [Far06; Mil01] and experimentally ascertained in Section 5.2.2, that HITS algorithm is badly behaved on certain kind of graphs, and therefore it will have non-unique and zero authority weights for important nodes.

Farahat et al., [Far06] developed Exponentiated Input method to address the limitations of HITS described above. HITS directly depends on adjacency matrix \( A \), which means in a given iteration step only paths of length 1 is considered. That is, to determine the authority score of a page \( p_i \), at each iteration HITS looks only at the hub scores of the adjacent pages (pages that directly point to page \( p_i \)) and similarly for hub score. In the Exponentiated HITS we consider paths of length greater than 1, by exploiting the exponentiation of matrix \( A \) in a controlled fashion. The matrix \( A \) is thus replaced with the “Taylor series” matrix:

\[
\begin{align*}
A + A^2/2! + A^3/3! + \ldots + A^m/m! + \ldots &= e^A - I \\
\end{align*}
\] (4.56)
The path of length 1 is considered more important than paths of length greater than 1 by using the scaling factor $1/m!$. Now the $I$ and $O$ operations of HITS can be re-written as:

\[
a_k = (e^A - I) h_{k-1} \tag{4.57}
\]

\[
h_k = (e^A - I) a_{k-1} \tag{4.58}
\]

Given a weakly connected\footnote{A directed graph $G$ is weakly connected if any node can be reached from any other node by traversing edges either in their indicated direction or in the opposite direction.} input graph, the Exponentiated HITS prevents both the problems identified above; see the proof on [Far06]. With the above adjustment now Exponentiated HITS can take into account paths of length greater than 1 in a given iteration.

Suppose,

\[
A_{EI} = (e^A - I)
\]

After having a proof of uniqueness and non-nil-weighting now with matrix $A_{EI}$ we are in a position to directly address personalization. The Exponentiated matrix $A_{EI}$ can be used as a step towards the personal-\[\text{alized} \text{ HITS algorithm. By incorporating user's personalization data into the base matrix } A, \text{ the subsequent Exponentiated matrix } A_{EI} \text{ will provide a better representation of the users’ personalization. With every power of } A \text{ the personalization data will get boosted and propagated. It requires a further in depth analyses to observe personalization together with Exponentiated Input. A possible future work will be to observe the calibrated effect of personalization due to different Exponentiated inputs. In the next chapter we will experiment with the Exponentiated Input in HITS together with personalization scheme similar to the PageRank model, also described in the next subsection.}

We can use also the link matrix $A'$ to hold the weights of connections between the pages based on usage data from webserver log of traffic. The link matrix can be initialized to 0, and then increment the link from node $i$ to $j$ every time a user travels from page $i$ to $j$. The effect of the matrix $A'$ would be similar to the gradient ascent method described in previous section. The link matrix now contains the usage information, and the entries in the matrix are larger when certain pages are visited more often. Thus in each iteration the most frequently followed links play larger role in determining new authority weights. The matrix $A'$ can be called as the usage weighted input matrix.

The usage logs provide a wealth of information that the relevancy ranking process can harness to improve search quality. How to traverse them, how to represent them in the link matrix and their expositions in personalization are quite motivating inquisitions in the field of usage log exploitation. Interpreting and utilizing these information is certainly not easy task. There has been some work done on exploiting usage or interaction information obtained from search engine logs, in order to understand how these implicit feedbacks (the logs) could be used to tune the relevancy ranking [Agi06a; Agi06b].

In Chapter 5 we will experiment with the personalized Exponentiated HITS and compare the results with other alternative algorithms, e.g., comparing the results with basic HITS and its improvements.

Randomized HITS

If link analysis is to provide a robust notion of authoritativeness in such a setting where information is dynamic in nature, it is natural to ask that is it also robust in the sense of being stable to perturbation of the link structure [Ng01b]. We can view the perturbations to network as continuous, abrupt and drastic changes in the adjacency matrix (changes in hyperlink structure). Perturbation analyses in linear algebra has very widespread
applications and implications. Through such analyses we would like to measure the consistency of the algorithms by exposing it to diverse and dynamic sets of input data.

The robustness of HITS algorithm like other link analysis algorithms requires an analysis of the stability of the eigenvector calculations. Issues like relationship between multiple eigenvectors and invariant subspaces, and the effect of random jumps must be considered [Ng01b; Ng01a]. In perturbation analysis, small changes to a matrix tend to result in large changes to eigenvectors only when eigenvalues are sufficiently close to each other [Mot95]. That is, when the eigengap\(^8\) of the matrix \(A^T A\) is large HITS is expected to be insensitive to small changes in the link structure of the matrix \(A\). But if the eigengap is small then slight changes in the link structure of matrix \(A\) will lead to large changes in ranking produced by HITS algorithm. Stability is therefore a desirable property of LAR algorithms, given the fact that the link structure of web changes quite frequently.

In the study by Ng et al., [Ng01a], they have experimentally found that HITS under certain conditions is quite instable, and also provided conditions under which HITS can be stable. They also found that PageRank algorithm is relatively immune to stability concerns. Ng et al., [Ng01b] later developed a modification to HITS algorithm, inspired by PageRank’s immunity to stability conditions. They considered a random surfer model similar to PageRank case, but in HITS the random surfer follows the hyperlinks both in the forward and backward directions, and occasionally “resets” and jumps to a page chosen uniformly at random. The method is named as “Randomized HITS”, because of the random surfer model. Mathematically, after the modification the I and O operations in HITS can be rewritten as:

\[
\begin{align*}
a^{(k+1)} &= \alpha \mathbf{1} + (1 - \alpha) A^T \mathbf{h}^{(k)} \\
h^{(k+1)} &= \alpha \mathbf{1} + (1 - \alpha) A \mathbf{a}^{(k+1)}
\end{align*}
\]

With the help of the above modifications, analogous to the PageRank, now the Randomized HITS is more stable to perturbations than the original basic HITS.

The Markov chain constructed as a result of the above two operations can be interpreted as: with probability \((1 - \alpha)\) the random surfer follows the hyperlink from the current page in forward direction when I operation is executed, and follows the hyperlink from current page in backward direction when O operation is executed. The random surfer teleports or jumps to a uniformly chosen destination with probability \(\alpha\).

The adjustment proposed in equations (4.59) and (4.60) bring in the damping factor in essentially the same way as that of PageRank. A consequence is that, we now have a room for personalization in HITS analogous to the PageRank model. In equations (4.59) and (4.60) if we manage to control the teleportation of the random surfer we could incorporate personalization. Notice that if we could replace the vector \(\mathbf{1}\) with non-uniform probability vector \(\mathbf{v}\), we will have the personalization in Randomized HITS, thus:

\[
\begin{align*}
a^{(k+1)} &= \alpha \mathbf{v} + (1 - \alpha) A^T \mathbf{h}^{(k)} \\
h^{(k+1)} &= \alpha \mathbf{v} + (1 - \alpha) A \mathbf{a}^{(k+1)}
\end{align*}
\]

With the above modification to the Randomized HITS, now the random surfer who teleports with probability \(\alpha\), will be enforced to jump to non-uniformly chosen destination based on the personalization vector \(\mathbf{v}\).

Motivated further by the above discussion a modification similar to the PageRank primitivity trick can also be applied to HITS as well. Making the HITS quite conclusive for the generic formulation of personalization given in Section 4.4.2. Thus mathematically the modified matrix can be written as:

\[
\alpha A^T A + (1 - \alpha) E
\]

\(^8\)The difference between largest and second largest eigenvalues of matrix \(A\), i.e., \(|\lambda_1 - \lambda_2|\), and for the matrix \(A^T A\) the eigengap will be \(|1 - \lambda_2|\).
With the above modification, the authority matrix becomes irreducible and therefore by *Perron-Frobenius theorem*, it possesses a unique and positive dominant eigenvector [Lan06; Kit98]. With the above modifications in equations (4.61), (4.62) and (4.63) we are now convinced that HITS can be both stabilized and personalized. We will experimentally ascertain the viability of these modifications in the next chapter.

Personalization is relatively new phenomenon in IR community, but the implications are quite futuristic. Therefore, there is a lot of scope available for further research in the field of personalization. The area is very fascinating and rewarding, and you have very widespread opportunities for manoeuvring. Personalization is employed also to improve performances apart from just personalizing the results, e.g., the BlockRank algorithm described in previous sections. Personalization can also be employed to control the search outcomes. It spans from the query subsystem to the relevancy subsystem in IR. In the query disambiguation process, personalization can be employed to obtain user specific information. It is therefore not far from truth that the prevalent and exclusive use of personalization in retrieval is the *current* and will be the *future* outlook of IR.

In the next chapter we will investigate further into empirical implications and results of the methods discussed in this chapter.
Experimental Evaluations

Experimental evaluation is to assess the quality of the algorithms and, more importantly to understand and observe how theoretically predicted properties manifest themselves in the practical setups. We will explore the practical exhibitions of ideas discussed in LAR algorithms earlier in this study.

In this chapter the focus is on the empirical evaluations of the algorithms discussed in Chapters 3 and 4. And also provide the practical manifestations of the key ideas discussed. We will specifically observe the effectiveness of the query-dependent LAR algorithms, such as; HITS, SALSA and their improvements. We will study their convergence behaviours and the significance of the modifications and improvements applied on them. Specifically we will observe the peculiarities of Extrapolation techniques in improving the rate of convergence in query-dependent algorithms.

The algorithms will be exposed against different sets of data. We have described the datasets that we will use for our experiments in next section, the experimental setup. Because of unavailability of sample and assessed datasets, we primarily rely on the dataset used in [Bor05]. We will thus compare the effectiveness of our findings with the findings described in the work by Borodin et al. We will also compare our findings with findings in both [Naj07b; Naj07a].

We will start with the experimental setup, and present the set of queries that will be used to test the algorithms. Those queries have already appeared in the previous researches [Cha98; Kle99; Bor05].

The overall focus of the chapter is to see the relationship between the theoretical properties of the query-dependent algorithms (defined earlier in Chapters 3 and 4) and their empirical outcomes.

5.1 Experimental Setup

5.1.1 The Graph and the Dataset

The algorithms that we will test, operate on a collection of pages that is created following the guidelines of Kleinberg (see Section 3.4). As is described in the work done in [Bor05] the search engine Google is queried for each of the queries shown in Table 5.1 (when a query consists of more than one word, we put the ‘+’ symbol in front, so as to ensure that all pages contain the query terms). The first 200 pages returned by Google form the Root-set as prescribed by Kleinberg. For each page in the Root-set, all the out-links are stored of that page, and the first 50 in-links, in the order they are returned by Google, as described in Section 3.4.3. One way of obtaining in-links is to use Google queries of form link : url (e.g., link : www.fastsearch.com), which returns
a list of documents that point to the url. The Root-set is then expanded into the Base-set by including the in-links, and out-links of the pages in the Root-set (see the graphical representation of this process in figure 3.4). Given the Base-set, the underlying graph is constructed, induced by this set of pages: each page is represented as a node, and a (directed) edge is placed for each link between the pages (nodes). Edges that connect two nodes within the same domain are removed since they usually serve navigation purposes, and isolated or dangling nodes are also deleted. The final graph will be given as input to the LAR algorithms.

The running time of creating the base-set graph is completely dominated by the time it takes to fetch the documents. It takes quite some time to download the documents, given that every neighbourhood graph contains nodes on the order of at least 2000+ documents.

The dataset that we will use in our experimentations has been taken from [Bor01]. On the prescription of Kleinberg as described in last two paragraphs, the dataset are stored in an inverted file format. Every page is first assigned a ‘docid’ (document id), and from the pages’ docids an inverted file will be generated corresponding to each query. An inverted file (for query “amusement parks”) might look like:

```
1    : 182, 183, 12, −1
    ;
5    : 325, 326, 327, 328, 329, −1
    ;
51   : 1296, 1297, 694, 707, 715, 789, 502, −1
    ;
129  : 2992, 3304, 2994, 3306, 3307, 3308, −1
130  : 1122, 3314, 3315, −1
    ;
3403 : 3405, 3406, −1
```

This means that docid 1 the first row (boldface), contains link to docids 182, 183 and 12, and −1 indicate end of list (out-links). Using the inverted file as an input to a script (such as, a bash-script, or a python code or a matlab code), we could convert the inverted file to an adjacency matrix A form. Where the entries of A corresponding to docid-1 will be; \( A(1, 182) = 1 \), \( A(1, 183) = 1 \) & \( A(1, 12) = 1 \) and rest of the entries in 1st row of matrix A will be; \( A(1, j) = 0 \), where \( j \notin (182, 183, 12) \). The resultant adjacency matrix A corresponding to each query given in Table 5.1, can be given as an input to the LAR algorithms.

5.1.2 The Queries

There are some standard set of queries appeared in the literature and used in previous works [Bor05; Kle99; Lem00]. The choices of queries are driven by the fact that they become a representative of the whole Web. Therefore it is expected that through their representativeness they unveil the implicit properties of the algorithms, e.g., HITS support of tightly knit communities (see section 3.4). Every query represents a topic on web. Webgraph can be considered as a set of clusters of strongly connected nodes, each cluster theoretically represents some topic(s). In principle we want to align query topic(s) with the topic(s) on the webgraph. By testing the algorithms with different representative queries (topics), we tend to observe the behaviour of the algorithms on these topic(s), using single or multi-topic queries at a time.

There are queries where the most relevant results are not textually expressed in the most relevant documents, e.g., the phrase “search engine” doesn’t appear in the most of the search engines main pages. Thus we would have those types of queries also which are usually not expressed within the relevant documents, such as, “search engines”, “automobile industries”, etc.
There are also queries for which we have conflicting communities on the webgraph. For example the query "iraq war" and "abortion" can have sets of conflicting clusters of webgraph. It is interesting to observe how different algorithms treat these queries.

The queries in the Table 5.1 with the statistical information will be used for the experiments; the queries are exactly the same as is used in [Bor05].

### 5.1.3 Query Statistics

The query statistics provided in the [Tsa04a] are used in our study (see Table 5.1). The table provided here shows the analytical study of the datasets corresponding to each query. The assessment information of the dataset gives a broad picture of the neighbourhood graph (the base-set). The information therefore is useful for conceptualizing and understanding the underlying structure of graph and broad picture of manifestation of
algorithms. Sometimes from the assessed dataset we could predict the expected behaviours, performances or outcomes of the algorithms. For example, for the query “search engines” there are 11,659 nodes and 292,236 links, which means the underlying graph for this query is quite big (11,659 \times 11,659) and dense. There will be memory contention issues for such a big graph. It would be interesting to observe how different algorithms will react to such a big graph.

In case of PageRank, Haveliwala [Hav99], presents a memory efficient approach that lowers the main memory requirements for the huge webgraph (using Block-Based strategy). In HITS usually the memory concern is not that terrible, because the graphs are usually of order 1,000 – 5,000 nodes. Nevertheless, efficient usage of memory is a favourable property for HITS too, given the fact that it is computed at query time, and the retrieved pages could be sizable.

Profoundly analyzed datasets provide valuable input for assessment or comparison of the LAR algorithms. Therefore it’s very crucial to have a sample and representative datasets with statistical information available, which could be used for experimentation. We could have done our own evaluation based on the pre-labelled corpus, such as TREC collection\(^1\), but due to limited time constraint, we primarily rely on the dataset provided on [Tsa].

We refer and confide on the query statistics given in the Table 5.1 during experimentations.

### 5.1.4 Measures

To assess the quality and accuracy of the results of an algorithm we will compare the ‘precision’ over top – 15 results with the results in [Bor05]. Usually for the relevancy ranking algorithms the measures used to assess the algorithms are ‘precision’ and ‘recall’. But we expect the result of the text based search to have high recall, and therefore we only consider the precision over top – 15 results, and also considering the behaviour web user, we can only consider the accuracy of the top – 15 results or the first page of results.

Given a query the dataset that we are using has been classified as non-relevant, relevant or highly relevant to the topic of the query (see [Tsa]). The classification is based on the relevance ratio of the query topic with page in question. Almost similar notion of relevance is employed in the TREC conference for topic distillation queries. In TREC the data relevance and high relevance is usually predefined by a set of experts.

### 5.1.5 Convergence

Convergence is the measure of how the ordering of the pages changes as the number of iterations increases. In general an algorithm should declare convergence once the value of Residual, \( R_{\text{Residual}, k} = \text{Rank}_{k+1} - \text{Rank}_k \) stabilizes. This means that the new iterations cannot change the ranking order, the ranking order has now stabilized.

In most of the literature \( L_1 \) norm or residual of the authority weights of two successive iterates is used to detect convergence in query-dependent LAR algorithms. For measuring the convergence in all of our algorithms we will therefore rely on the following measure:

\[
\delta_k = \| A \bar{x}^{(k)} - \bar{x}^{(k)} \|_p = 1 \tag{5.1}
\]

In linear algebra, equation (5.1) is generally used as an indicator of convergence for most of the iterative algorithms. In almost all of the experiments, \( L_1 \) norm in equation (5.1) is compared against \( \epsilon \in (10^{-16} - 10^{-5}) \). We will frequently refer to the variable \( \epsilon \), during the proceedings of this chapter, therefore it is wise to keep it in mind.

\(^1\)Text REtrieval Conference(TREC) is the primary benchmark for information retrieval.
There are other possible ways also to measure convergence, for instance in \cite{Kam03b} Kendall’s-tau rank correlation (KDist) measure is used, to see if the residual, $L_1$ norm is a good measure of convergence. Haveliwala \cite{Hav99} suggests to use induced orderings, rather than residuals, by looking at the ordering of the pages induced by the rank vector to measure convergence. With induced ordering, PageRank vector converges in as few as 10 iterations in comparison with convergence with $L_1$ residual which takes about 50+ iterations.

We choose $L_1$ norm as a measure of convergence for the sake of simplicity. But it’s a good idea to try different measures of convergence as described in the previous paragraph, to see their effectiveness on the number of iterations and the accuracy of rankings.

### 5.1.6 User Study

In the user study, a group of users have been shown the results of different algorithms (see \cite{Tsa}). The user study has been performed on the dataset that we are using. The results were permuted, so as that they appeared in random order, no information about the algorithm(s) was revealed to the users. The users ranked every document as, “Highly relevant”, “Relevant”, “Non-Relevant” or “Don’t know”. User feedback on the results produced by the algorithm is quite important and therefore provides information about the practical usefulness of different approaches. User feedback can be used to assess the quality of algorithms. The number of users feedbacks per query is shown in Table 5.2.

In \cite{Bor05} the users’ information has been collected online, for each of the algorithms they tested. Average is taken over all users and all queries on the feedbacks received. Considering the lack of information of users on some of the topics, their feedback introduced some noise too. But due to limited number of users per query the quantity of errors were measurable. On the user rating for a specific document, the document is rated as “Relevant” if the “Relevant” and “Highly Relevant” votes are more than “Non-Relevant” votes, but in case of ties “Non-Relevant” votes are favoured (pessimistic approach). For more information about the user study and the dataset see \cite{Bor05; Tsa}.

In the next sections the identified algorithms will be tested and evaluated in the light of experimental setups described in this section.

### 5.2 Algorithms and Results of Experiments

We have implemented most of the algorithms discussed in Chapter 4, to observe them in different experimental settings. In this section we will test them according to the experimental setups described in previous sections. The algorithms will be tested against the datasets corresponding to each of the 34 queries defined in Table 5.1. That is, the adjacency matrix $A$ corresponding to each query will be given as an input to the algorithms.

Broadly the following algorithms will be implemented:

- HITS (Basic, Extrapolated and Personalized)
- AT (Authority threshold - threshold set as average out-degree)
- Norm (Norm Family - with Euclidean Norm)
- HubAvg (Hub Average)
- Max (Max Operator)
- SALSA
<table>
<thead>
<tr>
<th>query</th>
<th>users</th>
</tr>
</thead>
<tbody>
<tr>
<td>abortion</td>
<td>22</td>
</tr>
<tr>
<td>affirmative action</td>
<td>7</td>
</tr>
<tr>
<td>alcohol</td>
<td>8</td>
</tr>
<tr>
<td>amusement parks</td>
<td>8</td>
</tr>
<tr>
<td>architecture</td>
<td>7</td>
</tr>
<tr>
<td>armstrong</td>
<td>8</td>
</tr>
<tr>
<td>automobile industries</td>
<td>7</td>
</tr>
<tr>
<td>basketball</td>
<td>12</td>
</tr>
<tr>
<td>blues</td>
<td>8</td>
</tr>
<tr>
<td>cheese</td>
<td>5</td>
</tr>
<tr>
<td>classical guitar</td>
<td>8</td>
</tr>
<tr>
<td>complexity</td>
<td>4</td>
</tr>
<tr>
<td>computational complexity</td>
<td>4</td>
</tr>
<tr>
<td>computational geometry</td>
<td>3</td>
</tr>
<tr>
<td>death penalty</td>
<td>9</td>
</tr>
<tr>
<td>genetic</td>
<td>7</td>
</tr>
<tr>
<td>geometry</td>
<td>7</td>
</tr>
<tr>
<td>globalization</td>
<td>5</td>
</tr>
<tr>
<td>gun control</td>
<td>7</td>
</tr>
<tr>
<td>iraq war</td>
<td>8</td>
</tr>
<tr>
<td>jaguar</td>
<td>5</td>
</tr>
<tr>
<td>jordan</td>
<td>4</td>
</tr>
<tr>
<td>moon landing</td>
<td>8</td>
</tr>
<tr>
<td>movies</td>
<td>10</td>
</tr>
<tr>
<td>national parks</td>
<td>6</td>
</tr>
<tr>
<td>net censorship</td>
<td>4</td>
</tr>
<tr>
<td>randomized algorithms</td>
<td>5</td>
</tr>
<tr>
<td>recipes</td>
<td>10</td>
</tr>
<tr>
<td>roswell</td>
<td>4</td>
</tr>
<tr>
<td>search engines</td>
<td>5</td>
</tr>
<tr>
<td>shakespeare</td>
<td>6</td>
</tr>
<tr>
<td>table tennis</td>
<td>6</td>
</tr>
<tr>
<td>vintage cars</td>
<td>5</td>
</tr>
<tr>
<td>weather</td>
<td>9</td>
</tr>
</tbody>
</table>

**average** | 7

*Table 5.2: Users per query*
5.2.1 Extrapolation

In this section we will assess the algorithms experimentally by comparing them with their Extrapolated version. As discussed in Section 4.3 the purpose of employing extrapolation is to understand convergence property more closely. Extrapolation therefore, is to exploit the convergence property and principally accelerate convergence by subtracting off estimates of the nonprincipal eigenvectors.

In this section we will exclusively and profoundly look into convergence property of the algorithms. The application of a single extrapolation step is considered to be equivalent to 0.5 times or less the cost of an iteration of power method (32% of cost of an iteration of power method [Havb]). Therefore the effects of extrapolation step is not that severe. We will interpret the improvements in number of iterations almost equivalently to the improvements in time, e.g., a 2 times improvement in number of iterations will be treated as 1.8 – 2.0 times speedup in the wall-clock time.

Specifically in this section we will make use of Quadratic Extrapolation (see Section 4.3.2). We will primarily show that the speedup due to Quadratic Extrapolation is on the scale of 30 – 900%, when we apply it carefully.
In a few of the experiments we even got a higher speedup than this scale (see Table 5.3), we will present them too during the proceedings.

The Extrapolated versions of each of the aforementioned algorithms are implemented according to the formulations defined in Section 4.3. We will take one algorithm at a time and analyze and compare its convergences in terms of how many times extrapolation step is applied, and how the algorithms respond to it. We will present the convergence trend in a graphical form. In the convergence graph the $y-axis$ represents the log of the $L_1$ norm (see equation (5.1)) of the two successive iterates, and $x-axis$ represents the number of iterations.

**HITS**

Recall HITS algorithm (see section 3.4), in this section we will compare experimentally HITS and Extrapolated HITS. The adjacency matrix $A$ (the base-set) corresponding to each query is used as an input to each of the algorithms. The adjacency matrix is constructed according to the prescription described in Section 5.1.1.

HITS will compute authority and hub vectors by iteratively reaching to a stable state by determining the dominant eigenvectors of the matrices $A^T A$ and $AA^T$. We will compare the convergence trend of the basic HITS with the Extrapolated HITS in numerous expositions.

![Figure 5.1: Convergence graph for query “alcohol”](image)

Starting off from the query “alcohol”, the adjacency matrix $A$ corresponding to the base-set (for query “alcohol”) is given as an input to HITS and Extrapolated HITS. The resultant curves portraying the rate of convergence are shown in the figure 5.1.

Clearly the convergence is significantly better in Extrapolated version of HITS. It reaches to the residual $\epsilon$ much quicker and with less number of steps than original HITS. We have tailored the parameters in Extrapolation to see any further gains or irregular behaviours. Applying extrapolation twice provides a better convergence
than applying it 5 times in this example. When extrapolation is applied twice we see convergence in 21st iterations while applying extrapolation 5 times the convergence occur after 22nd iterations.

The quadratic extrapolation technique should be applied periodically to subtract off the errors in the current iterate (along the direction of the second and third eigenvectors). It improves convergence only when it is applied carefully. It is interesting to observe empirically that extrapolation applied too frequently doesn’t really achieve any further benefits. Because by doing that we are not allowing the power method to use the new computed iterate to annihilate error components of the iterate in directions along the eigenvectors with small eigenvalues. Quadratic Extrapolation step leaves error components primarily along the smaller eigenvectors, which the power method is better equipped to eliminate. Thus we need to allow power method to eliminate errors instead of applying extrapolation step frequently and accumulating the errors after every application. That is why it is very much important to apply the right number of extrapolation steps to gain the required improvements.

In the convergence graph (figure 5.1) only extrapolation is applied twice, i.e., on 3rd and 7th iterations. Increase in the number of extrapolation step therefore doesn’t really offer any significant change in this example. So, it’s a matter of applying the sufficient number of extrapolations in different time periods in order to enhance the convergence of the future application of the power method.

Note that the ranking produced by both the Extrapolated HITS and original HITS are exactly the same. Extrapolation only improves convergence and the ranking order remains the same as in the original algorithm. It does not affect the ranking order at all.

Figure 5.2: Convergence for query “affirmative action”

In another example we consider a multi-topic query “affirmative action”. The normal HITS will take 2,529 iterations to reach $\epsilon$ residual. While the extrapolated HITS converge significantly faster in just 397 iterations – extrapolation is applied 9 times only. From the figure 5.2 we can visibly notice a considerable speedup in
Extrapolated HITS.

The spikes seen in the graph (see the blue curve in figure 5.2) are due to the acceleration step, but speedup occurs nevertheless. The spikes apparently seem as poor approximation due to the extrapolation, but immediately after the spike we see a very drastic drift in the convergence. We have seen this behaviour in most of the cases, i.e., there is a spike at acceleration step and immediately after the spike we see a very rapid drop in convergence (see all the convergence curves in Appendix A).

From the experiments with HITS it’s now quite evident that for extrapolation to be effective we have to apply it on the right time and the sufficient number of times during iterations. Sometimes applying late, sometimes applying early in iterations, sometimes applying more frequently and sometimes applying less frequently will give us incredible improvements in convergences, in an order of 100 – 900% (see also Appendix A and Table 5.3. It’s strongly encouraged that the reader goes through the results in appendices while reading the description here).

It requires further in depth analyses to optimize and thus automate the application of the extrapolation. That is, to read the properties of the convergence graph and also the behaviours of algorithms and figure out when and how many times extrapolation should be applied to get required improvements in convergence and the wall-clock time. If the convergence curve is already steep enough, applying extrapolation at that part of graph might not be helpful. But if the curve is mildly converging it might be very helpful to apply extrapolation step on that part of the convergence curve.

Figure 5.3: Convergence for query “death penalty”

Lack of good understanding of the extrapolation technique and convergence behaviour of algorithm might lead to a poor performance also due to extrapolation (see figure 5.3).

Extrapolation with HITS provides a substantial performance gain on the slow convergent sequences and
necessary insight about the capabilities of the extrapolation. We also observed that as the number of iterations increases we see more *noticeable* and *drastic* improvement in the Extrapolated HITS (see Appendix A).

**HubAvg (Hub Average)**

Recall the improvement of HITS, HubAvg (see Section 4.2.2), in this section we will compare HubAvg and *Extrapolated* HubAvg. Here also we intend to see any further outstanding variations in the convergence behaviour of algorithm due to extrapolation.

HubAvg generally performs quite good. The *top* − 15 results capture most of the authoritative sources (see Appendix B). As far the implementation is concerned it uses most of its time in calculating the average matrix, \( F \), see equation (4.1). This means that if the adjacency matrix \( A \) is large, the cost of computing the average matrix \( F \) might cause some delays. For example, for queries “search engines”, “weather”, “architecture”, “basketball” and “genetic”, it took quite some time to just compute \( F \). Because the size of adjacency matrices are in range \((6,000 \times 6,000) – (12,000 \times 12,000)\), hence it requires some computational time to operate on them and a huge memory to contain them. Hence there is a need to optimize the computation of average matrix \( F \) in HubAvg algorithm.

Extrapolation is reasonably effective in HubAvg too, in most of the cases the convergence speedup are even better than its predecessor HITS. The speedup due to extrapolation is observed to be of order \(150 – 930\%\), see for example convergence curves in figure 5.4.

![Figure 5.4: Convergence graph for query “computational complexity”](image)

In almost all the experiments with HubAvg, we see a sharp spike when the extrapolation step is applied (see Appendix A). But immediately after the spike we also observe sharp steepness in the curve, as in Extrapolated HITS.
Considering the case of the multi-topic query “computational complexity”, there is an astonishing speedup in convergence (see figure 5.4). In this case extrapolation is applied only 9 times, and each time it’s applied we see a drastic decline in the convergence curve. See the rate of change of residual in extrapolated HubAvg and normal HubAvg, visibly the rate at which the extrapolated HubAvg drive towards the solution is almost vertical in comparison with the normal algorithm.

We regulate the parameters by applying extrapolation either too often or seldom, early in iterations or late in iterations, in order to gain a better appreciation. We also intend to see the order of improvements in convergence due to extrapolation. We have established that applying extrapolation late when an uneven trend is found in convergence, and applying it frequently if each application of the extrapolation gives additional improvements. Thus applying extrapolation less frequent only if every application does not bring any radical improvements (see this behaviour in convergence curves in Appendix A).

The performance boost both in wall-clock time and convergence due to extrapolation is promising. The cost of extrapolation step is in order of 0.5 times a single iteration of the Power Method or even less, given the fact that we apply extrapolation only periodically (in the case of HubAvg, on an average we applied extrapolation 7 times, see Table 5.3), we can highlight that the extrapolation has quite minimal overhead but has quite impressive outcomes.

**Authority Threshold - AT (k)**

To reduce the effects of the weak authorities on the computation of the hub weight, while at the same time retain the positive effects of the strong authorities, Authority Threshold is applied to retain only the top few authorities. Therefore Authority threshold, AT (k), is to say that a node is good hub if it points to at least k good authorities (see Section 4.2.2).

In this subsection we will consider AT - Avg (taking the average out-degree from Table 5.1 as k), and compare it with its Extrapolated counterpart. Each iteration is computationally intense because computing the hub scores (∅ operation) entails accumulating k good authorities. A layman implementation of threshold-operator therefore doesn’t offer any computational lift. After SALSA, one of the algorithms which took a lot of time in experiments is AT-Avg. Unlike SALSA (which delays largely due to slow convergence), the delay here is mainly due to inefficiency of the ∅ operation.

It has been noticed (in AT-Avg) that the size of matrix A does not really influence the number of iterations required to converge. For example for the query “search engines”, with a big matrix of size (11,659 × 11,659) (see Table 5.1), the original AT-Avg took 40 iterations while the extrapolated version took 17 iterations (with 4 application of extrapolation step) to converge, see figure 5.5. Of course a big matrix can slow down the ∅ operation in AT-Avg, but it has negligible effects on the number of iterations.

Again applying extrapolation frequently or seldom and applying it earlier or late depends on the convergence trend of the algorithm. But the extrapolated AT-Avg’s performance is quite promising too, i.e., a performance gain of order 100 − 900%, see Table 5.3.

As the cost of a single iteration of AT-Avg is significant, therefore it’s very much important to have less number of iterations and hence a sharp reduction in the residuals is highly essential. But applying any other efficient method for the threshold-operator together with Extrapolation would have more promising results in AT-Avg in terms of wall-clock time.

See figure 5.6 for the query “abortion”, the extrapolated AT-Avg converges after just 25th iterations when extrapolation is applied only 6 times, while the normal AT-Avg algorithm converges after 171st iterations.
Figure 5.5: Convergence graph for query “search engines”

Figure 5.6: Convergence graph for query “abortion”
The irregular and steep convergence curve with spikes at the extrapolation point ascertains the theoretical essence of extrapolation, i.e., it subtract off approximation to the second and third eigenvectors, see Section 4.3.2, and therefore provide a much quicker convergence than normal.

Consider now the multi-topic query “vintage cars”, it takes only 49 iterations and extrapolation applied 4 times versus 273 iteration of the normal AT-Avg. The extrapolation steps are employed after 5\textsuperscript{th} iteration. As seen from the figure 5.7 and as expected, after every application of extrapolation there is a sharp reduction in $L_1$ Residual.

The observations about convergence due to extrapolation in the previous algorithms are also valid in the case of Authority threshold.

**Norm(2)**

Recall *Norm* family of algorithms from last chapter (see Section 4.2.2) – to scale the weights in a way so that lower authority weights contribute less to the hub weight. Hub weights of the nodes are taken to be the *p-norm* of the vectors of the authority weights. In the case of *Norm (2)*, it’s simply the vector norm or Euclidean norm (see equation (4.8)).

The efforts per iteration in *Norm (2)* is not that much in comparison with AT-Avg and HubAvg, but the improvement due to extrapolation is reasonably comparable. An improvement of order 12; (163 iterations by extrapolated version and 2136 by original *Norm (2)*), provides a very good reason to prefer extrapolation over the original Norm.

It should also be noticed (from Appendix B) that top – 15 results of the *Norm (2)* is the subset of the union of top – 15 results of HITS and HubAvg. The TKC effect in HITS appears to concur in both HubAvg and Norm (2) algorithms. Looking deeper into TKC effect is out of scope of this study, for detailed analyses on TKC effect see [Tsa04a; Bor05].
5.2 ALGORITHMS AND RESULTS OF EXPERIMENTS

(a) Convergence for query “recipes”

(b) Convergence for query “jordan”  

Figure 5.8: The convergence graphs - Norm(2) algorithm.  

Max

Max algorithm, see Section 4.2.2, a special case of both AT algorithm for threshold value $k = 1$ and of Norm $(p)$ algorithm for value $p = \infty$, i.e., $\infty - \text{norm}$. The hub score in $\odot$ operation is therefore computed by applying max operator to authority weights.

Like AT algorithm, Max algorithm is also computationally expensive per iteration because of the application of max operator. Max algorithm is deemed as second best algorithm in their study by Borodin et al., [Bor05]. To an extent Max algorithm gets rid of the TKC effect present in it predecessor HITS (see top – 15 results in Appendix B). It is also deemed to be good enough in finding relevant webpages to a webpage query. That is, Max algorithm is capable of discovering webpages, related to a query webpage [Tsa04b].

The improvement in convergence by extrapolated Max algorithm is quite analogous to that of Extrapolated AT-Avg and Norm algorithms.

For the query “basketball”, the extrapolated Max is 19 times (the most astonishing result) faster than the normal Max (see figure 5.9). Extrapolation applied just 6 times (after $5^{th}$ iteration), forced the slowly converging sequence to rapidly converge in just 42 iterations. Without extrapolation Max algorithm took 835 iterations to converge.

SALSA

As introduced by Lempel and Moran (see Section 3.5), we have also exposed SALSA to the experimental evaluation, in order to observe its convergence behaviour in exposition to Extrapolation. As with the previous algorithms, we tweak the parameters in the algorithm to sort out any observable conduct in the convergence curves.
From the experiments, SALSA takes a lot of iterations to converge, in average more than 1000 iterations to reach to residual $\epsilon$. In contrast to the findings by Najork [Naj07a], SALSA does not really outperforms HITS (or its descendants) in convergence behaviours. But in terms of relevancy, the top – 15 results produced by SALSA seem to resemble the antecedent of all LAR algorithms, the InDegree algorithm (see Section 3.2 and for results see Appendix B, and compare it with [Tsa]).

The worst case is when query “complexity” took 37,709 iterations to converge, while the extrapolated version took about 13,000 iterations (where extrapolation is applied 39 times) to reach to a $L_1$ residual $10^{-12}$. By changing the starting point and frequency of the extrapolation we get convergence after 4,353 iterations; extrapolation is applied 9 times starting after 1000th iterations. Thus it took half the number of iterations of the case when extrapolation is applied 39 times (see figure 5.10).

Careful application of extrapolation therefore gives a speedup of a factor of 2 over applying it haphazardly. Therefore it’s very much important to see when it is required and appropriate to apply extrapolation and how often it should be applied to gain a pragmatic and significant improvement.

Any other measure of convergence might be helpful; $L_1$ residual doesn’t really scale very well with SALSA. A similar measure as described by Haveliwala (see Section 5.1.5) applied to SALSA might provide any interesting insight about its convergence sensitivity. It might be that we donot need to iterate so much to converge to an almost good solution. Thus detecting convergence from induced ordering might be helpful in case of SALSA. If the ranking order stays the same irrespective of the weights (values), then applying any further iteration might not bring any significant changes except the slight changes in values (weights).
Summary and Results of Extrapolation

Extrapolation techniques are therefore very effective, the extrapolated algorithms in our experiments yielded a net speedup of over 3 (see Table 5.3), the speedup could be even more significant in practice; for example we even got a speedup of 19 on our dataset depending on careful application of extrapolation step (see figure 5.9(a) for query “basketball”).

See table 5.3 for a comprehensive overview of all the results for each of 34 queries defined in Table 5.1 that we used, and for all of the algorithms that we discussed in this section. In the table:

- \(itr\), is the number of iterations
- \(ext\), is the number of times extrapolation applied
- \(E\), is the Extrapolated version of the algorithms, and
- \(N\) refers to normal version

<table>
<thead>
<tr>
<th>Query</th>
<th>HITS</th>
<th>#</th>
<th>E</th>
<th>N</th>
<th>E</th>
<th>N</th>
<th>E</th>
<th>N</th>
<th>E</th>
<th>N</th>
<th>E</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>abortion</td>
<td>str</td>
<td>12</td>
<td>17</td>
<td>43</td>
<td>106</td>
<td>25</td>
<td>171</td>
<td>25</td>
<td>17</td>
<td>53</td>
<td>17</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>17</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>affirmative</td>
<td>str</td>
<td>397</td>
<td>25.29</td>
<td>71</td>
<td>199</td>
<td>79</td>
<td>327</td>
<td>125</td>
<td>356</td>
<td>75</td>
<td>182</td>
<td>375</td>
</tr>
<tr>
<td>action</td>
<td>ext</td>
<td>3</td>
<td>10</td>
<td>7</td>
<td>17</td>
<td>7</td>
<td>14</td>
<td>7</td>
<td>19</td>
<td>9</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>alcohol</td>
<td>str</td>
<td>21</td>
<td>38</td>
<td>41</td>
<td>135</td>
<td>17</td>
<td>35</td>
<td>23</td>
<td>42</td>
<td>21</td>
<td>35</td>
<td>281</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>18</td>
<td>4</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>amusement</td>
<td>str</td>
<td>26</td>
<td>97</td>
<td>44</td>
<td>61</td>
<td>37</td>
<td>86</td>
<td>95</td>
<td>194</td>
<td>47</td>
<td>81</td>
<td>648</td>
</tr>
<tr>
<td>parks</td>
<td>ext</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>24</td>
<td>10</td>
<td>9</td>
<td>24</td>
</tr>
<tr>
<td>architecture</td>
<td>str</td>
<td>45</td>
<td>98</td>
<td>153</td>
<td>817</td>
<td>109</td>
<td>792</td>
<td>48</td>
<td>131</td>
<td>41</td>
<td>164</td>
<td>1021</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>4</td>
<td>17</td>
<td>11</td>
<td>9</td>
<td>19</td>
<td>24</td>
<td>9</td>
<td>24</td>
<td>19</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>armstrong</td>
<td>str</td>
<td>30</td>
<td>68</td>
<td>26</td>
<td>44</td>
<td>29</td>
<td>33</td>
<td>32</td>
<td>51</td>
<td>61</td>
<td>502</td>
<td>6400</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>35</td>
<td>5</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>automobile</td>
<td>str</td>
<td>50</td>
<td>175</td>
<td>64</td>
<td>217</td>
<td>25</td>
<td>51</td>
<td>35</td>
<td>60</td>
<td>29</td>
<td>56</td>
<td>1171</td>
</tr>
<tr>
<td>industries</td>
<td>ext</td>
<td>9</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>25</td>
<td>5</td>
<td>25</td>
<td>5</td>
</tr>
</tbody>
</table>
### Table 5.3: Results of the experiments with Extrapolation

The extrapolation columns in the Table 5.3 indicate the best performance in terms of number of iterations that we got as a result of tweaking the parameters. Overall we have applied extrapolation steps 8 times on an average to get rapid convergences. So, the overhead of net application of extrapolation is very less, almost 3 – 4 iterations of power method. The last two columns in the Table 5.3 show the slow converging behaviours of SALSA (on average 7255 iterations). But the other columns on average converge after just 46 as a result of extrapolation in comparison to the average 170 iterations of the original algorithms. A net average speedup of

<table>
<thead>
<tr>
<th>Query</th>
<th>#</th>
<th>HITS</th>
<th>HubAvg</th>
<th>AT-avg</th>
<th>Norm(2)</th>
<th>Max</th>
<th>SALSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>basketball</td>
<td>irr</td>
<td>20</td>
<td>27</td>
<td>49</td>
<td>119</td>
<td>25</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>4</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>blues</td>
<td>irr</td>
<td>29</td>
<td>52</td>
<td>48</td>
<td>163</td>
<td>25</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>6</td>
<td>13</td>
<td>9</td>
<td>6</td>
<td>9</td>
<td>E</td>
</tr>
<tr>
<td>cheese</td>
<td>irr</td>
<td>17</td>
<td>24</td>
<td>44</td>
<td>87</td>
<td>46</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>classical guitar</td>
<td>irr</td>
<td>43</td>
<td>160</td>
<td>38</td>
<td>88</td>
<td>34</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>E</td>
</tr>
<tr>
<td>complexity</td>
<td>irr</td>
<td>21</td>
<td>32</td>
<td>101</td>
<td>321</td>
<td>26</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>3</td>
<td>11</td>
<td>5</td>
<td>5</td>
<td>9</td>
<td>E</td>
</tr>
<tr>
<td>comput. complexity</td>
<td>irr</td>
<td>53</td>
<td>144</td>
<td>64</td>
<td>661</td>
<td>18</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>comput. geometry</td>
<td>irr</td>
<td>28</td>
<td>58</td>
<td>44</td>
<td>98</td>
<td>34</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>death penalty</td>
<td>irr</td>
<td>12</td>
<td>18</td>
<td>40</td>
<td>70</td>
<td>14</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>genetic</td>
<td>irr</td>
<td>26</td>
<td>40</td>
<td>43</td>
<td>91</td>
<td>17</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>geometry</td>
<td>irr</td>
<td>25</td>
<td>45</td>
<td>45</td>
<td>108</td>
<td>23</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>E</td>
</tr>
<tr>
<td>globalization</td>
<td>irr</td>
<td>17</td>
<td>24</td>
<td>52</td>
<td>139</td>
<td>17</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>gun</td>
<td>irr</td>
<td>43</td>
<td>419</td>
<td>100</td>
<td>448</td>
<td>25</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>4</td>
<td>11</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>hash</td>
<td>irr</td>
<td>140</td>
<td>736</td>
<td>39</td>
<td>70</td>
<td>22</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>14</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>E</td>
</tr>
<tr>
<td>jaguar</td>
<td>irr</td>
<td>22</td>
<td>31</td>
<td>131</td>
<td>879</td>
<td>49</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>E</td>
</tr>
<tr>
<td>jordan</td>
<td>irr</td>
<td>30</td>
<td>78</td>
<td>262</td>
<td>131</td>
<td>62</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>moon landing</td>
<td>irr</td>
<td>34</td>
<td>79</td>
<td>59</td>
<td>178</td>
<td>25</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>movies</td>
<td>irr</td>
<td>131</td>
<td>568</td>
<td>41</td>
<td>72</td>
<td>21</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>national parks</td>
<td>irr</td>
<td>17</td>
<td>21</td>
<td>33</td>
<td>95</td>
<td>81</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>E</td>
</tr>
<tr>
<td>net censorship</td>
<td>irr</td>
<td>35</td>
<td>76</td>
<td>284</td>
<td>1048</td>
<td>18</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>3</td>
<td>18</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>E</td>
</tr>
<tr>
<td>randomized</td>
<td>irr</td>
<td>63</td>
<td>193</td>
<td>80</td>
<td>239</td>
<td>64</td>
<td>E</td>
</tr>
<tr>
<td>algorithms</td>
<td>ext</td>
<td>15</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>E</td>
</tr>
<tr>
<td>recipes</td>
<td>irr</td>
<td>90</td>
<td>397</td>
<td>53</td>
<td>113</td>
<td>28</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>12</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>roswell</td>
<td>irr</td>
<td>190</td>
<td>341</td>
<td>179</td>
<td>538</td>
<td>35</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>14</td>
<td>11</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>E</td>
</tr>
<tr>
<td>search engines</td>
<td>irr</td>
<td>9</td>
<td>13</td>
<td>25</td>
<td>41</td>
<td>17</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>E</td>
</tr>
<tr>
<td>shakespeare</td>
<td>irr</td>
<td>25</td>
<td>72</td>
<td>264</td>
<td>270</td>
<td>24</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>E</td>
</tr>
<tr>
<td>table tennis</td>
<td>irr</td>
<td>26</td>
<td>45</td>
<td>42</td>
<td>114</td>
<td>16</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>E</td>
</tr>
<tr>
<td>vintage cars</td>
<td>irr</td>
<td>35</td>
<td>60</td>
<td>91</td>
<td>587</td>
<td>49</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>E</td>
</tr>
<tr>
<td>weather</td>
<td>irr</td>
<td>26</td>
<td>55</td>
<td>32</td>
<td>54</td>
<td>17</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>ext</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>E</td>
</tr>
</tbody>
</table>

| Average             | irr | 50.2 | 191.7  | 67.9   | 247.1   | 33.6 | E     |
|                     | ext | 5.3  | 7.5    | 5.2    | 5.8     | 6.0  | E     |
| Median              | irr | 29.0 | 48.5   | 125.0  | 25.0    | 65.0 | E     |
|                     | ext | 4.0  | 2.0    | 5.0    | 5.9     | 6.0  | E     |
| StdDev              | irr | 69.1 | 443.3  | 52.0   | 263.8   | 21.9 | E     |
|                     | ext | 3.0  | 3.5    | 2.1    | 2.8     | 2.1  | E     |
order 5.78 in all the algorithms presents a very good reason for the usefulness of extrapolation techniques.

Note that Extrapolation technique can also be applied in conjunction with other acceleration techniques, such as BlockRank [Kam03a], or other iterative algorithms e.g., Gauss-Seidel, Successive Over Relaxation, Conjugate Gradient or any other methods [Gol96; Lan06]. When used in conjunction with any other methods, we might expect more insights about the effectiveness of Extrapolation, both in terms of time and convergence.

We have had a limited evaluation of the hybrid implementation of extrapolation technique (defined in Section 4.3.4). When we discuss the findings of Power $A^d$ Extrapolation we will also put forward the findings from hybrid approach of extrapolation. We have tested the application of power extrapolation interchangeable with quadratic extrapolation to discover any further speedup or insights.

### 5.2.2 Personalization

As explained in the last chapter (see Section 4.4) there has been some work done to personalize the search results according to the user’s internal model of “authority”, in this section we will provide empirical implications of some of those models. We will mostly experiment with the generic model for personalization described in Section 4.4.2. A slight effort towards modifying the weights of entries in the adjacency matrix, such as with Exponentiated HITS, in order to influence the authoritative sources.

In general, through personalization, we intend to boost certain interesting documents according to the user’s (or user group’s) preferences, usually represented by the personalization vector. Hence we will modify the original query-dependent LAR algorithms identified earlier, in order to incorporate them within personalization model. We will primarily expose HITS within the generic framework of personalization (see Section 4.4.2), and then present other possible ways for personalization, e.g., personalizing the Randomized and Exponentiated HITS.

The results produced by HITS and SALSA are maybe generally authoritative considering the network structure of pages, but may not reflect the user’s preferences. There is a valid rationale to modify basic HITS to take into account the user’s interests and therefore personalization is an effort towards changing the general notion of authority to user’s own and internal notion of authority and importance.

**Personalized HITS**

See equation (4.63) in the last chapter, according to the formulation $x = Qv$ in equation (4.54), we could also incorporate the same modification to the HITS equation (4.63). Thus the $I$ operation in HITS with personalization, will look like:

$$\vec{a}(k) = \alpha A^T A \vec{a}(k-1) + (1 - \alpha) \vec{v}$$  \hspace{1cm} (5.2)

where $\vec{v}$ is the personalized probability vector. According to the random surfer model, when random surfer gets bored, he jumps to randomly chosen destination according to the probability distribution given in $\vec{v}$. If $\vec{v}$ contains uniform probability distribution (e.g., $\vec{v} = \frac{1}{n}\vec{e}$), then the above equation will be the same as that of the original HITS. The convex combination of authority vector $\vec{a}$ and uniformly distributed personalized vector $\vec{v} = \frac{1}{n}\vec{e}$, experimentally gives the same results as the original HITS algorithm.

Consider the top – 15 result in Table 5.4 for query “search engines”, if we want to boost the page ‘About Web Search’ on 7th position (in boldface), we must change the uniformly distributed personalized vector $\vec{v}$ in equation (5.2) to a non-uniform probability vector. By giving a higher probability to the index 143 (representing the page index $P_{143}$) in the personalized vector $\vec{v}$, we are implicitly provoking the random surfer to choose


Table 5.4: Top 15 results for query “search engines”

<table>
<thead>
<tr>
<th>Rank</th>
<th>Page Index</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>P-135</td>
<td>AltaVista</td>
<td><a href="http://www.altavista.com">www.altavista.com</a></td>
</tr>
<tr>
<td>3.</td>
<td>P-5225</td>
<td>Yahoo! Danmark</td>
<td><a href="http://www.yahoo.dk">www.yahoo.dk</a></td>
</tr>
<tr>
<td>4.</td>
<td>P-3390</td>
<td>AltaVista Text-Only Search</td>
<td>ragingsearch.altavista.com</td>
</tr>
<tr>
<td>5.</td>
<td>P-4539</td>
<td>Euroseek</td>
<td>euroseek.net</td>
</tr>
<tr>
<td>6.</td>
<td>P-1721</td>
<td>Your Search Engine Internet directory, information, search engine</td>
<td><a href="http://www.searchpalm.com">www.searchpalm.com</a></td>
</tr>
<tr>
<td>7.</td>
<td>P-143</td>
<td>About Web Search - Guide to Search Engine Optimization &amp; Online S</td>
<td>websearch.about.com</td>
</tr>
<tr>
<td>8.</td>
<td>P-4979</td>
<td>Abacho - THE POWERFUL SEARCHENGINE!!</td>
<td><a href="http://www.abacho.co.uk">www.abacho.co.uk</a></td>
</tr>
<tr>
<td>10.</td>
<td>P-5766</td>
<td>Ananzi South Africa - Search Engine</td>
<td><a href="http://www.ananzi.co.za">www.ananzi.co.za</a></td>
</tr>
<tr>
<td>11.</td>
<td>P-1303</td>
<td>Empty title field</td>
<td><a href="http://www.portalhub.com">www.portalhub.com</a></td>
</tr>
<tr>
<td>12.</td>
<td>P-5283</td>
<td>DINO-Online Suchmaschine Webkatalog</td>
<td><a href="http://www.dino-online.de">www.dino-online.de</a></td>
</tr>
<tr>
<td>13.</td>
<td>P-5296</td>
<td>Lycos.de</td>
<td><a href="http://www.lycos.de">www.lycos.de</a></td>
</tr>
<tr>
<td>15.</td>
<td>P-5285</td>
<td>Fireball - Die Suchmaschine</td>
<td><a href="http://www.fireball.de">www.fireball.de</a></td>
</tr>
</tbody>
</table>

The top – 15 results in Table 5.5 shows that now the page P – 143 is moved from previous 7th position to the 4th position higher in the ranking. The damping factor is set to $\alpha = 0.6$ in this case. But in case if users like to see the interested page much higher, then we have to further reduce the damping factor. For example, if we have $\alpha = 0.15$, the resultant ranking order will be now given in Table 5.6. The other way to elevate page’s ranking is to raise it’s corresponding probability weight in the personalization vector $\vec{v}$.

If the page(s) to boost are lower down in the ranking, e.g., ‘WoYaa search engine’ appears on 38th position; boosting this page would require the random surfer to jump more often (i.e., $\alpha \leq 0.2$) and also the weight in the personalization vector should be relatively higher.

Hence it is highly rewarding to have an efficient and automated way, to orchestrate the values of the vector $\vec{v}$, the adjacency matrix $A$, and the damping factor $\alpha$. The automated solution can provide further insight about the effectiveness of these parameters on the ranking of the search outcomes. In our study we have manually

---

2African search engine and Website’s Directory
adjusted the parameters in order to observe their impact on ranking.

The other factor that is decisive in personalization, is the configuration of the personalized vector $\vec{v}$ for a particular user. The question is, how to accumulate and update the users specific personalized vectors $\vec{v}$, to capture or represent their interests. As discussed in the last chapter, the matrix $Q$, in the expression $x = Q \vec{v}$, provides the complete basis for personalized vector $\vec{v}$. But here we are only concerned with the one vector that represents a particular user’s preferences. There could be several ways to acquire the users’ preference information, for example, through relevance feedback cycles. Depending on the requirement and specification of the overall system, either each user or user group could be assigned individual personalization vector(s) representing their interests.

The personalization vector specific to a user or user group can be updated explicitly or implicitly as described in Section 4.4. Either they should be explicitly asked for their preferences, or their history (e.g., usage or interaction information from logs, such as click-through, time on page, etc) or bookmark information could be monitored and therefore could be used intelligently by the system to form their personalization vector. Acquiring meaningful implicit feedback from the tracing of the “noisy” users’ interaction information in search logs is yet another interesting area in personalization. Implicit relevance measures and feedbacks have been recently studied by several research groups and it’s an active and contemporary area in ranking and personalization [Agi06a; Agi06b].

User-profiles (capturing the implicit or explicit feedback) can be used to maintain the representation of the users’ interests. Considering the behaviour of web users, their interests and preferences keeps on changing repeatedly and unexpectedly. The IR system must keep track of the updates in the user preferences and hence make appropriate and necessary modifications in their assigned personalization vectors. In the study by Baudisch [Bau01], an information filtering architecture has been designed which is capable of dealing with frequently changing user interests. The architecture called as QuerySet filtering architecture (QSA) uses user-profiles to handle both gradual and abrupt changes in user preferences. Different interfaces are discussed through which user-aggregated relevance feedback are gathered by allowing them to manually handle major profile inaccuracies. For more information see [Bau01].

The next and main concern is the dissemination of user preferences and its generalization (as discussed in Section 4.4). Users do not really want to only boost the ranks of the specific pages that they consider

---

Table 5.6: Top 15 personalized results for query “search engines” ($\alpha = 0.15$)
authoritative (preferences), but they expect a *generalization* of their preferences. The set of authoritative pages that is maintained in the personalization vector (or user-profiles) should be used rather more actively than just to boost their rankings. This means that the system must not only raise the relevancy of the pages preferred by the user, but also boost the related pages to the preferred set. The authority given to the preferred pages must be extended to the related pages as well, which might be more authoritative. The idea of dissemination of user preferences is same as that of the *spread of activation* presented by Anderson and Pirolli [AND84].

As described in Section 4.4.4, through gradient ascent we could achieve the dissemination of authorities. The effect of gradient ascent would be quite similar to the approaches toward interpretation of user feedbacks from their interaction logs.

The empirical analyses for these approaches would be out of scope of this work. For further information on the approaches described here see [Mil01; Cha00; Pir96; Agi06b].

**Power Extrapolation (\(A^d\))**

As described in last chapter, there is another extrapolation method proposed by Haveliwala et al., (see Section 4.3.3). This method of extrapolation takes into account the second eigenvalue of the Markov matrix and uses it to subtract off the error along several non-principal eigenvectors.

There are numerous ways for finding the second eigenvalue of the Markov matrix (\(A^T A\) in case of HITS), as discussed in last chapter, *e.g.*, by deflation (which is computational prohibitive in our case). As described in Section 4.3.3, to get an approximation of the second eigenvalue of the row-stochastic Markov matrix \(A\), in the equation (4.45) the damping factor \(\alpha\) serve to be the second eigenvalue of the \(A\), i.e.:

\[
\lambda_2 = \alpha
\] (5.3)

The above expression is formally proved by Haveliwala and Kamvar in [Hava].

Therefore by using the damping factor as the second eigenvalue we could easily implement Power Extrapolation method. The purpose of evaluating power extrapolation under personalization is also because we will use the personalization scheme as a way to do the *power extrapolation*.

The convergence due to *Power Extrapolation* as observed by Haveliwala et al. [Havb], should be similar to *Quadratic extrapolation* evaluated in the last section. But with the query-dependent LAR algorithms, Power Extrapolation doesn’t manifest as desirable speedup as in Quadratic extrapolation case. We are getting on average improvements of order \(10-25\%\). Hence the application of Power Extrapolation doesn’t seem to be equivalent or comparable to that of Quadratic extrapolation in query-dependent algorithms.

Consider for example, the query “automobile industries” (see figure 5.11), the power extrapolated HITS converges in 88 iterations while normal HITS takes 104 iterations, the damping factor \(\alpha = 0.9\) and \(d = 6\) (recall \(A^d\)). Notice that extrapolation is only applied once. Also notice that the convergence curve of the power extrapolated HITS resembles the curve of the query independent PageRank shown in [Havb].

Note that in most of our experiments we kept the damping factor reasonably *high* in order to achieve better convergence. Thus, by doing that we are indirectly preventing the spammers to artificially inflict the rankings (recall the effect of the damping factor on spamming, Section 3.3.7).

The application of power extrapolation to query-dependent algorithms HITS and SALSA therefore does not offer any practically prominent improvements. But the application of power extrapolation is useful in a sense that it exploits the damping factor to accelerate the convergence. That is, it relies on already existing information and it’s quite simple to implement. The cost of one time application of power extrapolation is
5.2 ALGORITHMS AND RESULTS OF EXPERIMENTS

(a) Convergence for query “automobile industries”

(b) Convergence for query “computational complexity”

Figure 5.11: Convergence graphs - Power Extrapolated HITS algorithm.

almost negligible. And finally together with extrapolation, we now have a room for personalization as well with the help of power extrapolation.

Hybrid Extrapolation

The Hybrid approach discussed in Section 4.3.3 could be a novel approach towards extrapolation. We have applied Quadratic Extrapolation interchangeably with Power Extrapolation to see any abrupt trends in the convergences. But the convergences produced are almost comparable to the results in Quadratic Extrapolation. The limited evaluation of the hybrid doesn’t offer any significant insights into the usefulness of the hybrid approach, but the idea of exploiting multiple extrapolation techniques is well worth. The primary purpose of using the hybrid approach is to observe the inter-dependencies of the extrapolation techniques and the effect those dependencies on the convergence graph.

Consider for example, the multi-topic query “affirmative action”, we have applied Power Extrapolation step after 6th iteration, and after that Quadratic extrapolation took the charge. The convergence graph is shown in figure 5.12. There is a slight improvement in comparison to the figure 5.2, which was when we only applied Quadratic Extrapolation. In only Quadratic extrapolation approach we got convergence in 397th iteration while in the hybrid approach we are getting convergence in just 314th iteration. We have tested the hybrid approach with different settings, e.g., we have applied power extrapolation step at different time steps together with quadratic extrapolation. Sometimes applied at the start of the convergence curve, sometimes in the middle and sometimes at the end of the curve.

There is certainly more to the hybrid approach of extrapolation. The figure 5.12 provides a very cursory benefit of the hybrid approach of extrapolation, it could be influential in other cases. Instead of just using power and quadratic extrapolations, if any other new or existing extrapolation techniques could be used, it
might offer some more desirable improvements. It might as well be used for personalization more actively, e.g., now we could only boost certain pages through the help of power extrapolation. If the extrapolation premise or assumption (as defined in Section 4.3.4) is based on the personalization scheme then it could be possible to have twofold benefits of extrapolation, both in terms of convergence speedup and personalized outcomes.

**Exponentiated HITS**

Recall *Exponentiated HITS* (see Section 4.4.4), to realize the motivation we start with an example (see figure 5.13); running the original HITS algorithm first on the figure 5.13(a), yields the following authority and hub vectors:

\[
\begin{align*}
a & = (0, 0.1667, 0.1667, 0.0, 0.0, 0.0, 0.1667, 0.1667, 0.1667, 0.1667) \\
h & = (0.3333, 0.0, 0.0, 0.0, 0.0, 0.3333, 0.3333, 0, 0, 0, 0)
\end{align*}
\]

We see a lot of zeros and the weights are haphazardly repeated too, and hence the ranking appears to be impractical. Now running Exponentiated HITS on the same graph we get the following unique authority and hub vectors:

\[
\begin{align*}
a & = (0, 0.1029, 0.1029, 0.1048, 0.1048, 0.1062, 0.1062, 0.1062, 0.1062, 0.1062) \\
h & = (0.2382, 0.1237, 0.1237, 0.1298, 0.1298, 0.1298, 0.1274, 0, 0, 0, 0)
\end{align*}
\]

The Exponentiated HITS gives the following weights for the figure 5.13(b):

\[
\begin{align*}
a & = (0.2382, 0.1237, 0.1237, 0.1298, 0.1298, 0.1274, 0.1274, 0, 0, 0, 0) \\
h & = (0, 0.1029, 0.1029, 0.1048, 0.1048, 0.1062, 0.1062, 0.1062, 0.1062, 0.1062)
\end{align*}
\]

And the normal HITS:

\[
\begin{align*}
a & = (0.3333, 0.0, 0.0, 0.0, 0.3333, 0.3333, 0, 0, 0, 0, 0) \\
h & = (0, 0.1667, 0.1667, 0.0, 0.0, 0.0, 0.0, 0.1667, 0.1667, 0.1667, 0.1667)
\end{align*}
\]
Page 1, 6 and 7 have a tie in case of normal HITS on figure 5.13(b), while the Exponentiated HITS clearly provide a more practical ranking. Pages 2, 3, 4, and 5 have zero authority just like pages 8, 9, 10 and 11, on the same graph, which isn’t favourable.

From the above two examples in figure 5.13 we see that HITS actually doesn’t perform well on some types of graph, the rank produced are inconsistent (containing a lot of zero and repeated values). Exponentiated HITS in such types of situations provide a much practical relevancy scores by supplementing the inconsistencies in the basic HITS. Hence Exponentiated HITS provides authoritativeness more precisely and distinctively, consistent with the graph structure. Exponentiated HITS addresses both the repeated weights and zero weights problems existed in the original HITS.

After this motivation we are now ready to try out the personalized Exponentiated HITS. Like its predecessors Exponentiated HITS should be formulated within the general formulation of personalization.

On our given dataset corresponding to the queries in Table 5.1 Exponentiated HITS performs reasonably good. The top – 15 results depict an intersection of the authoritative sources in all the algorithms that we have discussed here. But the operation $(e^A - I)$ is computationally stiff, for a matrix $A$ of size $(5,000 \times 5,000)$, it takes about 2 – 5 minutes just to compute the exponentiation. But after exponentiation, the convergences are usually quite fast, on an average we had convergence in order 15 – 30 iterations.

The top – 15 results for the query “computational complexity” are shown in Table 5.7. The ranking order is different from HITS’s, but we see a lot of important authoritative pages in this list in comparison to the other algorithms (see Appendix B for comparison).
The Personalized Exponentiated HITS also offers twofold benefits – first that pages returned are more relevant and authoritative in general and secondly the possibility for personalization, like HITS.

For example, in the Table 5.7, in order to boost pages $P^{−241}$, $P^{−661}$ and $P^{−666}$, to appear in top−10 or higher in the ranking, we will apply Personalized implementation of Exponentiated HITS. Notice that as the interested documents are lower down in the ranking therefore random surfer has to teleport frequently, thus $\alpha = 0.05$. The results of personalization in Exponentiated HITS is shown in Table 5.8.

### Randomized HITS

Random surfer model as implemented in the previous sections and also in PageRank model. Here in the case of Randomized HITS (see Section 4.4.4) the primary purpose of random surfer model is to make HITS more stable to perturbations. Unlike the random surfer model in PageRank, the random surfer in Randomized HITS follows hyperlinks both in forward direction and in backward direction. The random surfer with guided teleportation will help to bias the rankings produced by Randomized HITS. In the original Randomized HITS model the jumps of the random surfer are chosen uniformly. But with the modifications in equations (4.61) and (4.62) the jumps of the random surfer can now be controlled with the personalization vector $\vec{v}$, in order to control the search outcomes. Hence this way we can direct the random surfer’s jumps based on the non-uniform probability distributions.

We have first implemented equations (4.59) and (4.60) and then tested it against the set of queries given in the Table 5.1. From the top−15 results on almost all queries, it is found that the Randomized HITS outcomes resemble the top−15 results of the primitive algorithm, the InDegree (described in Section 3.2). It is also observed that here the rate of convergence is quite rapid, on an average we get convergence in $10−20$ iterations.

The query “affirmative action” gives the top−15 results from running the Randomized HITS (see table 5.9). Now consider Table 5.10 the top − 15 results of InDegree Algorithm. Comparing them we visibly observe that the top − 15 results are ranked the same by the two algorithms. Randomized HITS hence resembles InDegree.
Table 5.8: Top 15 results for query “computational complexity”, Personalized Exponentiated HITS ($\alpha = 0.05$)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Page Index</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>P-2026</td>
<td>Copyright Information</td>
<td><a href="http://www.psu.edu/copyright.html">www.psu.edu/copyright.html</a></td>
</tr>
<tr>
<td>2.</td>
<td>P-1</td>
<td>Affirmative Action and Diversity Page</td>
<td>aad.english.ucsb.edu</td>
</tr>
<tr>
<td>5.</td>
<td>P-2</td>
<td>American Association for Affirmative Action</td>
<td><a href="http://www.affirmativeaction.org">www.affirmativeaction.org</a></td>
</tr>
<tr>
<td>6.</td>
<td>P-2381</td>
<td>Site Meter - Counter and Statistics Tracker</td>
<td><a href="http://www.cocounter.com/.../nnewman&amp;Lo=us">www.cocounter.com/.../nnewman&amp;Lo=us</a></td>
</tr>
<tr>
<td>7.</td>
<td>P-2382</td>
<td>Free web counter - Site access tracker - CQ Counter</td>
<td>sm6.sitemeter.com/stats.asp?siteid=sm6wobbly123</td>
</tr>
<tr>
<td>8.</td>
<td>P-2383</td>
<td>Free web counter - Site access tracker - CQ Counter</td>
<td><a href="http://www.now.org">www.now.org</a></td>
</tr>
<tr>
<td>10.</td>
<td>P-3</td>
<td>Affirmative Action Register</td>
<td><a href="http://www.eoaavt.edu">www.eoaavt.edu</a></td>
</tr>
<tr>
<td>11.</td>
<td>P-72</td>
<td>TEXT</td>
<td><a href="http://www.caasf.org">www.caasf.org</a></td>
</tr>
<tr>
<td>12.</td>
<td>P-1741</td>
<td>WIU - Division of Student Services</td>
<td>student.services.wiu.edu</td>
</tr>
<tr>
<td>13.</td>
<td>P-41</td>
<td>PSU Affirmative Action</td>
<td><a href="http://www.psu.edu/aaoffice">www.psu.edu/aaoffice</a></td>
</tr>
<tr>
<td>14.</td>
<td>P-904</td>
<td>The United States Department of Labor Home Page, Secretary of Lab</td>
<td><a href="http://www.dol.gov">www.dol.gov</a></td>
</tr>
</tbody>
</table>

Table 5.9: Top 15 results for query “affirmative action”, Randomized HITS
### Table 5.10: Top 15 results for query “affirmative action”, InDegree

<table>
<thead>
<tr>
<th>Rank</th>
<th>Page Index</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>P-2026</td>
<td>Copyright Information</td>
<td><a href="http://www.psu.edu/copyright.html">www.psu.edu/copyright.html</a></td>
</tr>
<tr>
<td>2.</td>
<td>P-1</td>
<td>Affirmative Action and Diversity Page</td>
<td>aad.english.ucsb.edu</td>
</tr>
<tr>
<td>5.</td>
<td>P-2</td>
<td>American Association for Affirmative Action</td>
<td><a href="http://www.affirmativeaction.org">www.affirmativeaction.org</a></td>
</tr>
<tr>
<td>7.</td>
<td>P-2382</td>
<td>Free web counter - Site access tracker - CQ Counter</td>
<td>cqcounter.com/.../id=nsnewman&amp;lo=us</td>
</tr>
<tr>
<td>8.</td>
<td>P-2383</td>
<td>Free web counter - Site access tracker - CQ Counter</td>
<td>cqcounter.com/.../id=nsnewman&amp;lo=us</td>
</tr>
<tr>
<td>10.</td>
<td>P-3</td>
<td>Affirmative Action Register</td>
<td><a href="http://www.aar-eeo.com">www.aar-eeo.com</a></td>
</tr>
<tr>
<td>11.</td>
<td>P-72</td>
<td>TEXT</td>
<td><a href="http://www.eoaavt.edu">www.eoaavt.edu</a></td>
</tr>
<tr>
<td>12.</td>
<td>P-7</td>
<td>CAA</td>
<td><a href="http://www.caasf.org">www.caasf.org</a></td>
</tr>
<tr>
<td>13.</td>
<td>P-1741</td>
<td>WIU - Division of Student Services</td>
<td>student.services.wiu.edu</td>
</tr>
<tr>
<td>14.</td>
<td>P-41</td>
<td>PSU Affirmative Action</td>
<td><a href="http://www.psu.edu/dept/aaoffice">www.psu.edu/dept/aaoffice</a></td>
</tr>
<tr>
<td>15.</td>
<td>P-904</td>
<td>The United States Department of Labor Home Page, Secretary of Lab</td>
<td><a href="http://www.dol.gov">www.dol.gov</a></td>
</tr>
</tbody>
</table>

### Table 5.11: Top 15 results for query “affirmative action”, Personalized Randomized HITS ($\alpha = 0.85$)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Page Index</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>P-2</td>
<td>American Association for Affirmative Action</td>
<td><a href="http://www.affirmativeaction.org">www.affirmativeaction.org</a></td>
</tr>
<tr>
<td>2.</td>
<td>P-3</td>
<td>Affirmative Action Register</td>
<td><a href="http://www.aar-eeo.com">www.aar-eeo.com</a></td>
</tr>
<tr>
<td>3.</td>
<td>P-2026</td>
<td>Copyright Information</td>
<td><a href="http://www.psu.edu/copyright.html">www.psu.edu/copyright.html</a></td>
</tr>
<tr>
<td>4.</td>
<td>P-1</td>
<td>Affirmative Action and Diversity Page</td>
<td>aad.english.ucsb.edu</td>
</tr>
<tr>
<td>5.</td>
<td>P-739</td>
<td>Adobe Acrobat Reader - Download</td>
<td><a href="http://www.adobe.com/.../readstep.html">www.adobe.com/.../readstep.html</a></td>
</tr>
<tr>
<td>7.</td>
<td>P-2381</td>
<td>Site Meter - Counter and Statistics Tracker</td>
<td>sm6.sitemeter.com/.../stats.asp?site=sm6wobbly123</td>
</tr>
<tr>
<td>8.</td>
<td>P-2382</td>
<td>Free web counter - Site access tracker - CQ Counter</td>
<td>cqcounter.com/.../id=nsnewman&amp;lo=us</td>
</tr>
<tr>
<td>9.</td>
<td>P-2383</td>
<td>Free web counter - Site access tracker - CQ Counter</td>
<td>cqcounter.com/.../id=nsnewman&amp;lo=us</td>
</tr>
<tr>
<td>11.</td>
<td>P-72</td>
<td>TEXT</td>
<td><a href="http://www.eoaavt.edu">www.eoaavt.edu</a></td>
</tr>
<tr>
<td>12.</td>
<td>P-7</td>
<td>CAA</td>
<td><a href="http://www.caasf.org">www.caasf.org</a></td>
</tr>
<tr>
<td>13.</td>
<td>P-1741</td>
<td>WIU - Division of Student Services</td>
<td>student.services.wiu.edu</td>
</tr>
<tr>
<td>14.</td>
<td>P-41</td>
<td>PSU Affirmative Action</td>
<td><a href="http://www.psu.edu/dept/aaoffice">www.psu.edu/dept/aaoffice</a></td>
</tr>
<tr>
<td>15.</td>
<td>P-904</td>
<td>The United States Department of Labor Home Page, Secretary of Lab</td>
<td><a href="http://www.dol.gov">www.dol.gov</a></td>
</tr>
</tbody>
</table>
algorithm. In almost all the 34 queries, randomized HITS produce a result quite similar to that of InDegree algorithm.

Now let’s observe the effects of Personalization on Randomized HITS, by implementing the equations (4.61) and (4.62). Hence this way we can include personalization vector \( \vec{v} \), which could be exploited to interleaver the outcomes.

For example, consider pages \( P-2 \) and \( P-3 \) in Table 5.9, for query “affirmative action”. To boost these two pages, and bring them in the top two positions we must appropriately alter the values in personalization vector \( \vec{v} \). Notice that \( \alpha \) here refers to teleportation probability. In this example setting \( \alpha = 0.85 \), which means 85% of time the random surfer teleports or jumps to new destination based on vector \( \vec{v} \), and 15% of time follows the hyperlinks in forward and backward directions based on the \( I \) and \( \Omega \) operations respectively. With these settings we have the personalized outcomes of randomized HITS shown in Table 5.11. Clearly the pages \( P-2 \) and \( P-3 \) now appear in top - 2 positions.

We haven’t observed Randomized HITS in relation to the perturbations. We therefore rely on the findings from [Ng01b] in case of stability, but personalization in Randomized HITS performs as theoretically predicted in Section 4.4.4.

**Summary and Reflection on Personalization**

The results of the experiments due to personalization portray very broad and general implications of personalization in query-dependent LAR algorithms. It requires a further in depth study, independently conducted with focus on personalization. The active integration of personalization information in the link matrices can be quite a fascinating extension. For example, by using the entries in the link matrix as weights representing the personalization information, possibly utilize the user interaction or usage information from logs in the link matrix to personalize the results based on user behaviour. In our study we could only manage to have an overview of these concepts, and a quite cursory experimental analysis.

There is certainly more to personalization than just boosting a couple of pages. As discussed in Section 4.4, we could employ personalization for other purposes such as anti-spamming and having a control on the set of relevant outcomes. The experiments that we conducted in personalization are aimed to see whether we can control the outcomes to an extent. And we found that it sufficiently possible to use personalization in the query-dependent algorithms without any major barriers. While exploring personalization we have also talked about various capabilities and similarities of different algorithms.

Of course there could more to be done with the ideas that are explored empirically and theoretically in this chapter. In the next chapter we will present the possible extensions and future work out of this study. We will also discuss the broad implications of this study.
Part III

Conclusions, Recommendations & Future Study
Chapter 6

Conclusions

IR is a very wide area; spanning all types and kinds of data and information retrieval. From its broad and wide range applications in databases and image processing, to more specific and active applications in multimedia retrieval and search engines, IR research has been around from decades to mesmerize these key areas. All sorts of dynamic and core concepts in mathematics, informatics and other relevant fields have been applied in IR research to make a difference.

Search based on pure content analysis of documents in the huge network of web was not enough. The reasons could be computational limitations on how to process such a huge amount of diverse and chaotic content, but also because of the innate limitations in content analysis; its incapability to provide good match to the user interests. Therefore there was a dire need to take the next step towards better understanding the match between the users’ interests and retrieval processes. Somehow extract or mimic the user behavioural information and use that as a valuable source of information for tuning and improving retrieval. This was the stimulus which invoked the efforts towards the active use of the latent information like ‘citation structure’.

Link Analysis Ranking was the next step from just content-analyses. It involves analyses and understanding of a very huge and jumbled network(s) of documents. From such a huge and massive network extracting useful information is quite a challenging and difficult task. The challenge is not just because of size of network, but also because of its diversity and unpredictability. The huge network(s) of documents hence forms the core of link analysis ranking.

Not only the challenge is due to huge network but also the interests of the users of IR have become very unpredictable. The actual challenge therefore is not just to understand or classify the huge network of documents but also to understand users’ changing needs and provide them a good match from highly dynamic sources of information.

In this study we had extensively explored IR problem from a general perspective and particularly went into certain depth on the topic of Link Analysis Ranking. We have discussed different perspectives in several contexts, which subsequently formed a strong basis for this research.

The goals from the general exploration of IR models were to have a strong standing and background of the problem in IR. We have explored wide varieties of novel approaches towards IR, for example, the use of Singular Value Decomposition (SVD) in latent semantic indexing (LSI) and HITS; the idea neural networks and inference networks. SVD, LSI and inference networks therefore formed a realistic motivation towards the focus of our study the Link Analysis Ranking.

In general, the task of digesting, conceptualizing and reasoning from a wide-range plethora of ideas and
concepts within IR is a difficult task in itself. After acquiring background knowledge in IR, figuring out any peculiarity, in the form of improvements of old ideas or discovery of new ideas is yet another challenge. Overall the topic of IR is a very difficult topic to deal with, but at the same very interesting, dynamic and rewarding topic as well.

6.1 Objectives of the study

The objectives of the study primarily were to explore and evaluate different models in IR particularly with a focus on relevancy ranking strategies. The following objectives were in mind when we started the study:

- To study document ranking strategies. To have an overview and understanding of the different concepts and ideas available in IR and their mathematical and logical fundaments.
- To come up with some radical improvements and evaluation of the existing models or propose some new strategy or model.
- To experimentally evaluate those models, verify the theoretical implications with empirical manifestations.

In this section we will take each of the objectives and relate it to the outcomes of this study, to see if the study outcomes are in line with the objectives.

6.1.1 Document ranking strategies

From the start of the study we kept our scope quite wide and flexible. We have started with the general formulation of the IR process in order to gather bits and pieces to be able to perceive the whole picture. By doing that it was intended to explore the whole spectrum of relevancy in IR; from classical content-analysis based models to vector based models, and probabilistic models to inference and neural network based models and ultimately to the hyperlinked and citation based models. The objective was to relate or classify them in a way to generalize the concepts in order to reuse or innovate them in some other situations.

We came across a lot of original and provocative models which used central concepts from mathematics and informatics. A lot of different mathematical notions are applied in a variety of situations. There is an extensive application of Boolean algebra, vector algebra, linear algebra, probability theory, artificial intelligence and other interesting areas of knowledge.

In link analysis ranking there is an extensive use of the core and theoretic notions of linear algebra. The use of Markov chain theory, Singular Value Decomposition, Optimization theory and sophisticated techniques from graph theory are few of the most used and highly talked about concepts in LAR. We have explored different ideas to certain comprehendible levels, which enabled to propose some fruitful contributions in our limited study.

6.1.2 Improvements and Contributions

After having a formal background of IR in general and link analysis ranking in particular, we then had to perform closer examinations of the models. During the analysis and evaluation we have discussed strengths, weaknesses and possible prospects of the different approaches. This way we were able to explore the implicit properties of numerous algorithms and exploit those properties to enhance their strengths and trim down their weaknesses.
Within the link analysis ranking we have primarily focused on query-dependent approaches for relevancy ranking. Query-dependent approaches usually have a direct consequence on the interaction time between user and system. Therefore it is highly appreciated to optimize or accelerate certain processes in the system in order to prevent any kind of delays or overheads at query-time. Our major contribution therefore is the improvements primarily in the convergence behaviour of the query-dependent algorithms. Speedup in convergence by reducing the number of iterations directly improves the interaction time.

Extrapolations are simple and unique techniques that require little additional infrastructure that needs to be incorporated in the existing LAR algorithms. We have distinctively applied extrapolation techniques to query-dependent LAR algorithms such as HITS and SALSA. We have manipulated the parameters extensively and extract a very novel performance gain due to Extrapolation (see Appendix A). By periodic application of extrapolation we have exceedingly enhanced the rate of convergence of the query-dependent LAR algorithms.

In the study by Kamvar et al., [Kam03b] they found an improvement of order 3 at-most due to extrapolation, in PageRank algorithm. By applying extrapolation much more carefully in the query-dependent algorithms, we have had improvements of order in range \((3 - 19)\), see Appendix A.

We have also incorporated personalization in the query-dependent algorithms. There is wide range of studies on personalization in IR research communities. We went through a handful of approaches towards personalization. We have formed a generic formulation for personalization, based on the random surfer model of PageRank. We have casted the query-dependent LAR algorithms in the generic framework of personalization. There were few of the query-dependent algorithms which had an intrinsic support for personalization, only it needed to be used in that context (the idea of random jumps).

6.1.3 Experiments and Evaluations

We have experimented with the algorithms that we proposed, as well as the ones that we studied. We have performed an extensive experimental study (see Appendices A and B). The purpose of the widespread experimental evaluation was primarily to observe the effects of the changes and improvements on the overall algorithm, and consequently to examine the manifestation of theoretical predicted properties.

For experiments we used a standard set of queries, based on their representativeness and on how they invoke some of the intrinsic properties and problems in LAR algorithms (e.g., TKC effect in HITS). The dataset corresponding the standard query set were previously used and analyzed by Borodin et al. [Bor05]. For each of the queries we had a neighbourhood graph (the base-set as prescribed by Kleinberg, see Section 3.4), which can be formed from an index structure with hyperlink information. The adjacency matrix corresponding to the base-set can be given as an input to the algorithms.

Setting up the experimental environment is the first major step, and hence not an easy task. The difficulty primarily is due to the availability of a representative set of data, and secondly the formations of an appropriate adjacency matrix that does not contain non-informative links and only contains informative links. And thirdly a computational capability is also required to test the algorithms in a realistic environment.

We have extensively experimented with the extrapolated and normal algorithms, and compared them with each other to see their responses to the improvements due to extrapolation. We therefore observed their convergence behaviour quite closely (see Appendix A).

The speedup in convergence due to extrapolation came first as a surprise. There were some interesting observations that came out as a result of the experiments. We observed that a careful application of extrapolation can improve convergence inevitably. Therefore, it matters when, where and how many times during iterations
you apply extrapolation step to gain the required acceleration. The importance of extrapolation in accelerating the convergence is hence remarkable. We have also tested hybrid extrapolation technique, where we observed the effect of extrapolation based on different techniques applied together.

We have experimented with personalization as well. Tested different algorithms and experimented with different personalization vectors. The query-dependent algorithms within the generic framework of personalization have been incorporated and observed. We have boosted a set of documents with help of personalization and observed their effects on the top − 15 results. We also felt the need for more sophisticated approaches for personalization, such as, the spread of authority from the boosted document(s) to the related documents, and an active incorporation of usage information in relevancy ranking model.

As a result of experiments we also found some resemblance between few of the algorithms as well. For example we discovered that the Randomized HITS resembles the simpler and primitive algorithm, the InDegree algorithm.

6.2 Future Work

Despite the already huge amount of available research on IR and relevancy ranking, there are still a lot of interesting areas open for more in depth exploration. We have also identified quite a few of the possible future work during the evaluations and analyses. There is still a lot of work to be done within Link Analysis Ranking. There were numerous things that we touched upon during this study which also require a further in depth research.

Analyses and observations of Graph Structure of Web is one of the most primitive and motivational areas in Link Analysis ranking. There is already extensive work done in the area of pure graph analysis for LAR research. But there is still a greater need to do more close analyses of the complicated network of documents, the webgraph. Because the webgraph is changing at an incredible rate, the mere addition or removal of information on web at any moment should be enough to motivate more focus on this area.

The existing researches formulated an overall structure of the web, e.g., the “bow tie” formulation, and the discovery of power law distribution. But a study totally based on the clustering of the graph, in terms of both finding out the good clusters and bad clusters, or a characterization of the good and bad clusters in webgraph can be very exemplary future work.

Extrapolation techniques can be further extended, by exploring other interesting ideas which could be used to either further improve the existing work or become a new entry. As a result of our study, it is possible to read much more closely the convergence graph from another perspective. For example, use any other convergence measure instead of just $L_1$ norm. The question is from the convergence behaviours, is it possible to automate when, where and how many times extrapolation should be applied to gain certain acceleration. Also the more active use of induced ordering [Hav99] to measure convergence together with extrapolation could be a possible future work. Hybrid approach of extrapolation could also be further observed to formulate a better framework for extrapolation, which could possibly be used for personalization too, apart from just accelerating the convergences.

Users’ usage data in the webserver logs is currently used as a heuristic for personalization and other purposes, for example, for upgrading the dictionaries with new keywords. But its rigorous use as weights in the link matrix could be appealing towards more active use of personalization.

IR can be thought of as a flow problem (law of conservation). We can make use of the pattern of traffic (e.g., from webserver logs), and the paths that users implicitly traverse in the graph, which means we need to
exploit information gained from traffic pattern on the web traffic. A possible future work could be to have a traffic model formulation of IR, which has a direct implication on personalization.

Personalization therefore is quiet central and contemporary topic in the research communities in IR. There is a need that one read more in depth about the spreading activation of the authoritative sources which is important in users’ perspective. Not just boost the documents in question but also spread or propagate the authoritativeness/importance among the related documents. The notion of authority must therefore be shifted from system to user.

Use of structural information of document can be beneficial in some places but can also be the requirement in another place. Structured information can be thought in realistic sense (i.e., chapters, sections, subsections), but it can also be thought in other settings. A nested structure can be artificially created for the documents (structured or un-structured) in the collection, in order to personalize or improve the retrieval process. A possible future work therefore could be to consider documents in nested structured approach.

### 6.3 Problems and impediments

In order to get a good theoretical background in IR you have to spend quite some time. There is a huge amount of work done in IR which spread across many different and diverse disciplines. The varieties of areas that we have discussed here in this study are just a very few of them. There is so much width and depth in IR problem space, at times it becomes difficult to just go through already existing work in any particular area. You would never reach to the bottom of any area in IR to have a comprehensive overview. It’s better to get contended at certain depth during the study.

For experimental setup and evaluations of the algorithms it’s quite hard to find out a sufficiently representative set of data. The data used should be assessed, analyzed and a good representative of the actual environment. This means you would confine your analyses on the representativeness and statistical information of the dataset. And if the dataset is not properly chosen and analyzed, the findings of the study might contradict in practice. And it’s usually true that algorithms that perform quite well on the experimental environment might not perform as expected in the realistic setup. Therefore it is increasingly important to choose a dataset quite carefully.

There are some road blockers and impediments on the way, and to overcome those blockers you really need to put some firm efforts. Therefore it’s quite necessary to be aware of the complexity of the problem before engaging in a formal study.

### 6.4 Recommendations and Broader Implications

A good background in the knowledge areas like Linear Algebra, Numerical analysis, Combinatorial Optimizations, Compiler construction (especially the lexical analyzer, for understanding the text operations), finite automaton, and also knowledge of Concurrent systems can be valuable before engaging in formal research in IR. A good background in Artificial intelligence, especially in neural networks and expert systems could be a great support for further studies.

This study is just a very broad and exterior overview of the IR models. It is very worthy to come within the spectrum of IR; primarily because the area is quite wide, your interests won’t remain unfulfilled in such wide range gamut of possibilities. Introducing new ideas are usually considered as valuable as improving the old ones in IR. There is always a room for new ideas to be used in this spectrum.
The task of studying IR is inevitably difficult but the difficulty also entails within it a wholesome of interests, rewards and desire for exploration. You never feel that you have nothing to do at any instance. Every new article and every new book that you read, you get more motivations and more understandings, and hence a stronger background. Writing this last part of thesis I still have to read some more articles. In the end, it has never been unfulfilled neither un-inspirational to study the Information Retrieval (IR).
Bibliography


BIBLIOGRAPHY


Experiments - Extrapolation
Figure A.1: Convergence graphs for query “abortion”
Figure A.2: Convergence graphs for query “affirmative action”
Figure A.3: Convergence graphs for query “alcohol”
(a) HITS
(b) HubAvg
(c) AT-Avg
(d) Norm (2)
(e) Max
(f) SALSA

Figure A.4: Convergence graphs for query “amusement parks”
Figure A.5: Convergence graphs for query “architecture”
Figure A.6: Convergence graphs for query “armstrong”
Figure A.7: Convergence graphs for query “automobile industries”
Figure A.8: Convergence graphs for query “basketball”
Figure A.9: Convergence graphs for query “blues”
Figure A.10: Convergence graphs for query “cheese”
Figure A.11: Convergence graphs for query “classical guitar”
Figure A.12: Convergence graphs for query “complexity”
Figure A.13: Convergence graphs for query “computational complexity”
Figure A.14: Convergence graphs for query “computational geometry”
(a) HITS
(b) HubAvg
(c) AT-Avg
(d) Norm (2)
(e) Max
(f) SALSA

Figure A.15: Convergence graphs for query “death penalty”
Figure A.16: Convergence graphs for query “genetic”
Figure A.17: Convergence graphs for query “geometry”
Figure A.18: Convergence graphs for query “globalization”
Figure A.19: Convergence graphs for query “gun control”
Figure A.20: Convergence graphs for query “iraq war”
Figure A.21: Convergence graphs for query “jaguar”
Figure A.22: Convergence graphs for query “jordan”
Figure A.23: Convergence graphs for query “moon landing”
Figure A.24: Convergence graphs for query "movies"
Figure A.25: Convergence graphs for query “national parks”
Figure A.26: Convergence graphs for query “net censorship”
Figure A.27: Convergence graphs for query “randomized algorithms”
Figure A.28: Convergence graphs for query “recipes”
Figure A.29: *Convergence graphs for query “roswell”*
Figure A.30: Convergence graphs for query “search engines”
Figure A.31: Convergence graphs for query “shakespeare”
Figure A.32: Convergence graphs for query “table tennis”
EXPERIMENTS - EXTRAPOLATION

Figure A.33: Convergence graphs for query “vintage cars”
Figure A.34: Convergence graphs for query “weather”
Appendix B

Experiments - Top-15 Results
Table B.1: Top 15 results for query “abortion”
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Welcome to Penn State's Home on the Web</td>
<td><a href="http://www.psu.edu/dept/aaoffice">http://www.psu.edu/dept/aaoffice</a></td>
</tr>
<tr>
<td>2</td>
<td>PSU Affirmative Action</td>
<td><a href="http://aad.english.ucsb.edu">http://aad.english.ucsb.edu</a></td>
</tr>
<tr>
<td>3</td>
<td>Welcome to Penn State's Home on the Web</td>
<td><a href="http://aad.english.ucsb.edu">http://aad.english.ucsb.edu</a></td>
</tr>
<tr>
<td>4</td>
<td>UC Berkeley home page</td>
<td><a href="http://www.berkeley.edu">http://www.berkeley.edu</a></td>
</tr>
<tr>
<td>5</td>
<td>University of Illinois</td>
<td><a href="http://www.purdue.edu">http://www.purdue.edu</a></td>
</tr>
<tr>
<td>6</td>
<td>Welcome to the University of Oregon Home Page</td>
<td><a href="http://www.uoregon.edu">http://www.uoregon.edu</a></td>
</tr>
<tr>
<td>7</td>
<td>The University of Arizona</td>
<td><a href="http://www.arizona.edu">http://www.arizona.edu</a></td>
</tr>
<tr>
<td>8</td>
<td>The University of Arizona</td>
<td><a href="http://www.arizona.edu">http://www.arizona.edu</a></td>
</tr>
<tr>
<td>9</td>
<td>The University of Iowa Home Homepage</td>
<td><a href="http://www.uiowa.edu">http://www.uiowa.edu</a></td>
</tr>
<tr>
<td>10</td>
<td>Welcome to Ohio University</td>
<td><a href="http://www.ohio.edu">http://www.ohio.edu</a></td>
</tr>
<tr>
<td>11</td>
<td>Welcome to the University of Vermont</td>
<td><a href="http://www.upenn.edu">http://www.upenn.edu</a></td>
</tr>
<tr>
<td>12</td>
<td>Welcome to the University of Vermont</td>
<td><a href="http://www.upenn.edu">http://www.upenn.edu</a></td>
</tr>
<tr>
<td>13</td>
<td>Welcome to the University of Vermont</td>
<td><a href="http://www.upenn.edu">http://www.upenn.edu</a></td>
</tr>
<tr>
<td>14</td>
<td>Welcome to the University of Vermont</td>
<td><a href="http://www.upenn.edu">http://www.upenn.edu</a></td>
</tr>
<tr>
<td>15</td>
<td>Welcome to the University of Vermont</td>
<td><a href="http://www.upenn.edu">http://www.upenn.edu</a></td>
</tr>
</tbody>
</table>

Table B.2: Top 15 results for query “affirmative action”
Table B.3: Top 15 results for query “alcohol”
Table B.4: Top 15 results for query “amusement parks”
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
<th>Hits</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All Conferences - Com</td>
<td><a href="http://www.allconferences.com">http://www.allconferences.com</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Castles of the World Tours</td>
<td><a href="http://www.castlesoftheworld.com">http://www.castlesoftheworld.com</a></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.castlesoftheworld.com">www.castlesoftheworld.com</a></td>
<td><a href="http://www.castlesoftheworld.com">http://www.castlesoftheworld.com</a></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Castle Tours: 12% Discount on Trafalgar Tour Prices - Trafalga</td>
<td><a href="http://www.affordableTrafalgartours.com">http://www.affordableTrafalgartours.com</a></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Contiki Tours: 5% discount on Contiki Tours</td>
<td><a href="http://www.affordableContikutours.com">http://www.affordableContikutours.com</a></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Collette Vacations: 5% discount on Collette Vacations</td>
<td><a href="http://www.affordablecolletelevisiontours.com">http://www.affordablecolletelevisiontours.com</a></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Ireland Tours: 5% Off CIE Tours.</td>
<td><a href="http://www.affordableIrelandtours.com">http://www.affordableIrelandtours.com</a></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Collette Vacations: 5% discount on Collette Vacations</td>
<td><a href="http://www.affordablecolletelevisiontours.com">http://www.affordablecolletelevisiontours.com</a></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Affordable Tours - Discounts on: Globus Tours, Trafalgar Tours, C</td>
<td><a href="http://www.affordableTours.com">http://www.affordableTours.com</a></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Affordable Tours - Discounts on: Celebrity Cruises, Crystal Cruises</td>
<td><a href="http://www.affordablecrusertours.com">http://www.affordablecrusertours.com</a></td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Affordable Budget Tours - Discounts on Celebrity Cruises, Crystal Cruises</td>
<td><a href="http://www.affordablecrusertours.com">http://www.affordablecrusertours.com</a></td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

**Table B.5: Top 15 results for query “architecture”**
Table B.6: Top 15 results for query “armstrong”
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
<th>Results 1</th>
<th>Results 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel - India Travel,Tourism In India,Travel to India</td>
<td><a href="http://www.indiatripportal.com">http://www.indiatripportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Empty title field</td>
<td><a href="http://www.bombaymotor.com">http://www.bombaymotor.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Empty title field</td>
<td><a href="http://www.indiatripportal.com">http://www.indiatripportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Empty title field</td>
<td><a href="http://www.indiantravelportal.com">http://www.indiantravelportal.com</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The table represents the Top 15 results for the query "automobile industries."
<table>
<thead>
<tr>
<th>Rank</th>
<th>URL</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.studentadvantage.com">http://www.studentadvantage.com</a></td>
<td>Student Advantage Discount Card for college students</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.unc.edu">http://www.unc.edu</a></td>
<td>Welcome to the University of North Carolina - Chapel Hill</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.mediascanlink.com/owu/unc">http://www.mediascanlink.com/owu/unc</a></td>
<td>University of North Carolina - Owego University - Official Athletic Info</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://g.msn.com/download/AK/08">http://g.msn.com/download/AK/08</a></td>
<td>MSN Search - More Useful Everyday</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://mail.und.com/email/scripts/useragreement.pl">http://mail.und.com/email/scripts/useragreement.pl</a></td>
<td>Your Fighting Irish E-mail!</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://www.inform.umd.edu/CampusInfo/Departments/Athletics/OAC">http://www.inform.umd.edu/CampusInfo/Departments/Athletics/OAC</a></td>
<td>University of Maryland Terrapin Club</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://g.msn.com/download/AK/10">http://g.msn.com/download/AK/10</a></td>
<td>MSN Money - More Useful Everyday</td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.studentadvantage.com">http://www.studentadvantage.com</a></td>
<td>Student Advantage Discount Card for college students</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.dukestore.com">http://www.dukestore.com</a></td>
<td>Student Advantage Discount Card for college students</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.fsu.edu">http://www.fsu.edu</a></td>
<td>Welcome to Florida State University - Seminole Boosters</td>
</tr>
<tr>
<td>11</td>
<td><a href="http://g.msn.com/download/AK/00">http://g.msn.com/download/AK/00</a></td>
<td>Welcome to MSN.com</td>
</tr>
<tr>
<td>12</td>
<td><a href="http://www.mediascanlink.com/owu/und">http://www.mediascanlink.com/owu/und</a></td>
<td>University of North Carolina - University of Maryland</td>
</tr>
<tr>
<td>13</td>
<td><a href="http://g.msn.com/download/AK/04">http://g.msn.com/download/AK/04</a></td>
<td>MSN Money - More Useful Everyday</td>
</tr>
<tr>
<td>14</td>
<td><a href="http://www.terrapinclub.com">http://www.terrapinclub.com</a></td>
<td>Welcome to the University of Maryland Terrapin Club</td>
</tr>
<tr>
<td>15</td>
<td><a href="http://www.studentadvantage.com">http://www.studentadvantage.com</a></td>
<td>Student Advantage Discount Card for college students</td>
</tr>
</tbody>
</table>

Table B.8: Top 15 results for query “basketball”
Table B.9: Top 15 results for query “blues”

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
<th>Hits</th>
<th>Page</th>
<th>Avg</th>
<th>Top-15</th>
<th>Tdfa Avg</th>
<th>Tdfa Hits</th>
<th>Tdfa Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Harry’s Blues Lyrics Online, Home Page, US</td>
<td><a href="http://www.harrysblueslyrics.com">http://www.harrysblueslyrics.com</a></td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>The Blues Foundation. Your Home for Blues, Blues, Blues!</td>
<td><a href="http://www.blues.org">http://www.blues.org</a></td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>The Blue Highway</td>
<td><a href="http://www.thebluehighway.com">http://www.thebluehighway.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>Delta Blues - DeltaBlues - deltablues.com</td>
<td><a href="http://www.deltablues.com">http://www.deltablues.com</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>51</td>
<td>51</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
Table B.10: Top 15 results for query “cheese”
Table B.11: Top 15 results for query “classical guitar”
Table B.13: Top 15 results for query “computational complexity”
Table B.16: Top 15 results for query “genetic”
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>The Geometry Center Welcome Page</td>
<td><a href="http://www.geom.umn.edu">http://www.geom.umn.edu</a></td>
<td>17</td>
<td>The Geometry Center Welcome Page</td>
<td><a href="http://www.geom.umn.edu">http://www.geom.umn.edu</a></td>
</tr>
</tbody>
</table>

**Table B.17:** Top 15 results for query “geometry”
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INDYMEDIA TIJUANA :: centro de medios independientes</td>
<td><a href="http://www.tijuanaimc.org">http://www.tijuanaimc.org</a></td>
</tr>
<tr>
<td>2</td>
<td>Independent Media Center - SALSA</td>
<td><a href="http://www.indymedia.org.nz">http://www.indymedia.org.nz</a></td>
</tr>
<tr>
<td>3</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.melbourne.indymedia.org">http://www.melbourne.indymedia.org</a></td>
</tr>
<tr>
<td>5</td>
<td>South Africa Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>6</td>
<td>Adelaide indymedia - webcast news - page 1</td>
<td><a href="http://www.indymedia.org">http://www.indymedia.org</a></td>
</tr>
<tr>
<td>7</td>
<td>Independent Media Center - Empty title field</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>8</td>
<td>Indymedia news - Aotearoa Independent Media Centre (AIMC)</td>
<td><a href="http://www.indymedia.org.nz">http://www.indymedia.org.nz</a></td>
</tr>
<tr>
<td>9</td>
<td>Urbana-Champaign Independent Media Center</td>
<td><a href="http://www.indymedia.org">http://www.indymedia.org</a></td>
</tr>
<tr>
<td>10</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>11</td>
<td>Vaikuttava Tietotoimisto (VAI)</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>12</td>
<td>North Texas Independent Media Center - Reclaim the Media</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>13</td>
<td>South Africa Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>14</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>15</td>
<td>Indymedia Italia - network di media indipendenti</td>
<td><a href="http://italy.indymedia.org">http://italy.indymedia.org</a></td>
</tr>
<tr>
<td></td>
<td><strong>AT-Avg</strong></td>
<td><strong><a href="http://www.indymedia.org">http://www.indymedia.org</a></strong></td>
</tr>
<tr>
<td>1</td>
<td>INDYMEDIA TIJUANA :: centro de medios independientes</td>
<td><a href="http://www.tijuanaimc.org">http://www.tijuanaimc.org</a></td>
</tr>
<tr>
<td>2</td>
<td>Independent Media Center - SALSA</td>
<td><a href="http://www.indymedia.org.nz">http://www.indymedia.org.nz</a></td>
</tr>
<tr>
<td>3</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.melbourne.indymedia.org">http://www.melbourne.indymedia.org</a></td>
</tr>
<tr>
<td>4</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://baltimoreimc.org">http://baltimoreimc.org</a></td>
</tr>
<tr>
<td>5</td>
<td>South Africa Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>6</td>
<td>Adelaide indymedia - webcast news - page 1</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>7</td>
<td>Indymedia news - Aotearoa Independent Media Centre (AIMC)</td>
<td><a href="http://www.indymedia.org.nz">http://www.indymedia.org.nz</a></td>
</tr>
<tr>
<td>8</td>
<td>Urbana-Champaign Independent Media Center</td>
<td><a href="http://www.indymedia.org">http://www.indymedia.org</a></td>
</tr>
<tr>
<td>9</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>10</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>11</td>
<td>Vaikuttava Tietotoimisto (VAI)</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>12</td>
<td>North Texas Independent Media Center - Reclaim the Media</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>13</td>
<td>South Africa Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>14</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>15</td>
<td>Indymedia Italia - network di media indipendenti</td>
<td><a href="http://italy.indymedia.org">http://italy.indymedia.org</a></td>
</tr>
<tr>
<td></td>
<td><strong>SALSA</strong></td>
<td><strong><a href="http://www.indymedia.org">http://www.indymedia.org</a></strong></td>
</tr>
<tr>
<td>1</td>
<td>INDYMEDIA TIJUANA :: centro de medios independientes</td>
<td><a href="http://www.tijuanaimc.org">http://www.tijuanaimc.org</a></td>
</tr>
<tr>
<td>2</td>
<td>Independent Media Center - SALSA</td>
<td><a href="http://www.indymedia.org.nz">http://www.indymedia.org.nz</a></td>
</tr>
<tr>
<td>3</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.melbourne.indymedia.org">http://www.melbourne.indymedia.org</a></td>
</tr>
<tr>
<td>4</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://baltimoreimc.org">http://baltimoreimc.org</a></td>
</tr>
<tr>
<td>5</td>
<td>South Africa Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>6</td>
<td>Adelaide indymedia - webcast news - page 1</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>7</td>
<td>Indymedia news - Aotearoa Independent Media Centre (AIMC)</td>
<td><a href="http://www.indymedia.org.nz">http://www.indymedia.org.nz</a></td>
</tr>
<tr>
<td>8</td>
<td>Urbana-Champaign Independent Media Center</td>
<td><a href="http://www.indymedia.org">http://www.indymedia.org</a></td>
</tr>
<tr>
<td>9</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>10</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>11</td>
<td>Vaikuttava Tietotoimisto (VAI)</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>12</td>
<td>North Texas Independent Media Center - Reclaim the Media</td>
<td><a href="http://adelaide.indymedia.org">http://adelaide.indymedia.org</a></td>
</tr>
<tr>
<td>13</td>
<td>South Africa Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>14</td>
<td>Melbourne Independent Media Center</td>
<td><a href="http://www.indymedia.org.au">http://www.indymedia.org.au</a></td>
</tr>
<tr>
<td>15</td>
<td>Indymedia Italia - network di media indipendenti</td>
<td><a href="http://italy.indymedia.org">http://italy.indymedia.org</a></td>
</tr>
</tbody>
</table>

Table B.18: Top 15 results for query "globalization"
### Table B.19: Top 15 results for query “gun control”

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Rifle Association - MyNRA</td>
<td><a href="http://www.nra.org">URL</a></td>
</tr>
<tr>
<td>2</td>
<td>Jews for the Preservation of Firearms Ownership</td>
<td><a href="http://www.gunowners.org">URL</a></td>
</tr>
<tr>
<td>3</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>4</td>
<td>The Violence Policy Center</td>
<td><a href="http://www.vpc.org">URL</a></td>
</tr>
<tr>
<td>5</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>6</td>
<td>Women Against Gun Control is a free to join group that supports</td>
<td><a href="http://www.gunfree.org">URL</a></td>
</tr>
<tr>
<td>7</td>
<td>Coalition to Stop Gun Violence/Educational Fund to Stop Gun Violence</td>
<td><a href="http://www.atf.treas.gov">URL</a></td>
</tr>
<tr>
<td>8</td>
<td>The Violence Policy Center</td>
<td><a href="http://www.vpc.org">URL</a></td>
</tr>
<tr>
<td>9</td>
<td>Women Against Gun Control is a free to join group that supports</td>
<td><a href="http://www.gunfree.org">URL</a></td>
</tr>
<tr>
<td>10</td>
<td>Keep and Bear Arms - Gun Owners Home Page - 2nd Amendment</td>
<td><a href="http://www.keepandbeararms.com">URL</a></td>
</tr>
<tr>
<td>11</td>
<td>Second Amendment Sisters', Inc.</td>
<td><a href="http://www.sas-aim.org">URL</a></td>
</tr>
<tr>
<td>12</td>
<td>American Freedom Foundation</td>
<td><a href="http://www.americanfreedomfoundation.org">URL</a></td>
</tr>
<tr>
<td>13</td>
<td>Coalition to Stop Gun Violence/Educational Fund to Stop Gun Violence</td>
<td><a href="http://www.atf.treas.gov">URL</a></td>
</tr>
<tr>
<td>14</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>15</td>
<td>Keep and Bear Arms - Gun Owners Home Page - 2nd Amendment</td>
<td><a href="http://www.keepandbeararms.com">URL</a></td>
</tr>
</tbody>
</table>

### At-Avg

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Rifle Association - MyNRA</td>
<td><a href="http://www.nra.org">URL</a></td>
</tr>
<tr>
<td>2</td>
<td>Jews for the Preservation of Firearms Ownership</td>
<td><a href="http://www.gunowners.org">URL</a></td>
</tr>
<tr>
<td>3</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>4</td>
<td>The Violence Policy Center</td>
<td><a href="http://www.vpc.org">URL</a></td>
</tr>
<tr>
<td>5</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>6</td>
<td>Women Against Gun Control is a free to join group that supports</td>
<td><a href="http://www.gunfree.org">URL</a></td>
</tr>
<tr>
<td>7</td>
<td>Coalition to Stop Gun Violence/Educational Fund to Stop Gun Violence</td>
<td><a href="http://www.atf.treas.gov">URL</a></td>
</tr>
<tr>
<td>8</td>
<td>The Violence Policy Center</td>
<td><a href="http://www.vpc.org">URL</a></td>
</tr>
<tr>
<td>9</td>
<td>Women Against Gun Control is a free to join group that supports</td>
<td><a href="http://www.gunfree.org">URL</a></td>
</tr>
<tr>
<td>10</td>
<td>Keep and Bear Arms - Gun Owners Home Page - 2nd Amendment</td>
<td><a href="http://www.keepandbeararms.com">URL</a></td>
</tr>
<tr>
<td>11</td>
<td>Second Amendment Sisters', Inc.</td>
<td><a href="http://www.sas-aim.org">URL</a></td>
</tr>
<tr>
<td>12</td>
<td>American Freedom Foundation</td>
<td><a href="http://www.americanfreedomfoundation.org">URL</a></td>
</tr>
<tr>
<td>13</td>
<td>Coalition to Stop Gun Violence/Educational Fund to Stop Gun Violence</td>
<td><a href="http://www.atf.treas.gov">URL</a></td>
</tr>
<tr>
<td>14</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>15</td>
<td>Keep and Bear Arms - Gun Owners Home Page - 2nd Amendment</td>
<td><a href="http://www.keepandbeararms.com">URL</a></td>
</tr>
</tbody>
</table>

### Hits

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Rifle Association - MyNRA</td>
<td><a href="http://www.nra.org">URL</a></td>
</tr>
<tr>
<td>2</td>
<td>Jews for the Preservation of Firearms Ownership</td>
<td><a href="http://www.gunowners.org">URL</a></td>
</tr>
<tr>
<td>3</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>4</td>
<td>The Violence Policy Center</td>
<td><a href="http://www.vpc.org">URL</a></td>
</tr>
<tr>
<td>5</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>6</td>
<td>Women Against Gun Control is a free to join group that supports</td>
<td><a href="http://www.gunfree.org">URL</a></td>
</tr>
<tr>
<td>7</td>
<td>Coalition to Stop Gun Violence/Educational Fund to Stop Gun Violence</td>
<td><a href="http://www.atf.treas.gov">URL</a></td>
</tr>
<tr>
<td>8</td>
<td>The Violence Policy Center</td>
<td><a href="http://www.vpc.org">URL</a></td>
</tr>
<tr>
<td>9</td>
<td>Women Against Gun Control is a free to join group that supports</td>
<td><a href="http://www.gunfree.org">URL</a></td>
</tr>
<tr>
<td>10</td>
<td>Keep and Bear Arms - Gun Owners Home Page - 2nd Amendment</td>
<td><a href="http://www.keepandbeararms.com">URL</a></td>
</tr>
<tr>
<td>11</td>
<td>Second Amendment Sisters', Inc.</td>
<td><a href="http://www.sas-aim.org">URL</a></td>
</tr>
<tr>
<td>12</td>
<td>American Freedom Foundation</td>
<td><a href="http://www.americanfreedomfoundation.org">URL</a></td>
</tr>
<tr>
<td>13</td>
<td>Coalition to Stop Gun Violence/Educational Fund to Stop Gun Violence</td>
<td><a href="http://www.atf.treas.gov">URL</a></td>
</tr>
<tr>
<td>14</td>
<td>The Million Mom March united with the Brady Campaign to Prevent Gun</td>
<td><a href="http://www.millionmommarch.org">URL</a></td>
</tr>
<tr>
<td>15</td>
<td>Keep and Bear Arms - Gun Owners Home Page - 2nd Amendment</td>
<td><a href="http://www.keepandbeararms.com">URL</a></td>
</tr>
</tbody>
</table>
Table B.20: Top 15 results for query “iraq war”
Table B.21: Top 15 results for query “jaguar”

<table>
<thead>
<tr>
<th>Rank</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.apple.com/games/trailers/ww2online">http://www.apple.com/games/trailers/ww2online</a></td>
</tr>
</tbody>
</table>
Table B.22: Top 15 results for query “jordan”
Table B.23: Top 15 results for query “moon landing”

<table>
<thead>
<tr>
<th>Rank</th>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.longdistanceratefinder.com">http://www.longdistanceratefinder.com</a></td>
<td>Long Distance Rate Finder - Best telephone calling services</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.longdistanceratefinder.com">http://www.longdistanceratefinder.com</a></td>
<td>Cognigen: Worldwide Telecommunications Long Distance</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.billzilla.com/apl">http://www.billzilla.com/apl</a></td>
<td>Dish Network Satellite TV</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://cognigen.net/talkamerica/apl">http://cognigen.net/talkamerica/apl</a></td>
<td>OneStar Communications - Long Distance And Local Telephone</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.cognidial.com/dial-around/apl">http://www.cognidial.com/dial-around/apl</a></td>
<td>ShopAETV.com - Product Detail</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://www.qksrv.net/click-310374-35140">http://www.qksrv.net/click-310374-35140</a></td>
<td>Name a Star - International Star Registry (R)</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.cognigen.net/onestar/apl">http://www.cognigen.net/onestar/apl</a></td>
<td>Exchange-It - Free Banner Exchange</td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.qksrv.net/click-310374-35140">http://www.qksrv.net/click-310374-35140</a></td>
<td>DISH Network e-Store</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.longdistanceratefinder.com">http://www.longdistanceratefinder.com</a></td>
<td>Cognigen: Worldwide Telecommunications Long Distance</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.billzilla.com/apl">http://www.billzilla.com/apl</a></td>
<td>DISH Network Satellite TV</td>
</tr>
<tr>
<td>11</td>
<td><a href="http://cognigen.net/talkamerica/apl">http://cognigen.net/talkamerica/apl</a></td>
<td>OneStar Communications - Long Distance And Local Telephone</td>
</tr>
<tr>
<td>12</td>
<td><a href="http://www.cognidial.com/dial-around/apl">http://www.cognidial.com/dial-around/apl</a></td>
<td>ShopAETV.com - Product Detail</td>
</tr>
<tr>
<td>13</td>
<td><a href="http://www.qksrv.net/click-310374-35140">http://www.qksrv.net/click-310374-35140</a></td>
<td>Name a Star - International Star Registry (R)</td>
</tr>
<tr>
<td>14</td>
<td><a href="http://www.cognigen.net/onestar/apl">http://www.cognigen.net/onestar/apl</a></td>
<td>Exchange-It - Free Banner Exchange</td>
</tr>
<tr>
<td>15</td>
<td><a href="http://www.qksrv.net/click-310374-35140">http://www.qksrv.net/click-310374-35140</a></td>
<td>DISH Network e-Store</td>
</tr>
</tbody>
</table>

Notes:
- The Long Distance Rate Finder is a website that offers a variety of telephone calling services. It includes a feature where you can name a star through the International Star Registry. The website also provides a feature to exchange your banner on other websites.
- Dish Network Satellite TV offers a variety of satellite TV services. It is a popular service in the United States and other parts of the world.
- OneStar Communications offers both long distance and local telephone services. It is an established company with a good reputation in the telecommunications industry.
- ShopAETV.com is an online store that sells various products, including electronics and home goods.
- Name a Star - International Star Registry offers a service where you can name a star through a reputable organization.
- Exchange-It is a free banner exchange service that allows you to exchange your banner on other websites.
- DISH Network e-Store is the official online store for DISH Network, offering a variety of products and services related to satellite TV.

For more information, please visit the websites listed above.
<table>
<thead>
<tr>
<th>Rank</th>
<th>URL</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><a href="http://www.classicfilm.about.com">http://www.classicfilm.about.com</a></td>
<td>American Film Institute</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.google.com">http://www.google.com</a></td>
<td>Google</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.allmovie.com">http://www.allmovie.com</a></td>
<td>The Internet Movie Database (IMDb)</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.ajax.com">http://www.ajax.com</a></td>
<td>The Internet &amp; TV Networks from Jupitermedia Corp.</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://www.signs.movies.com">http://www.signs.movies.com</a></td>
<td>Signs on DVD</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.imdb.com">http://www.imdb.com</a></td>
<td>The Internet Movie Database (IMDb)</td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.getwild.com">http://www.getwild.com</a></td>
<td>Get Wild - GetWild - getwild.com</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.filmsite.org">http://www.filmsite.org</a></td>
<td>Movie Review Query Engine</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.filmsite.org">http://www.filmsite.org</a></td>
<td>Movie Review Query Engine</td>
</tr>
<tr>
<td>11</td>
<td><a href="http://www.amazon.com">http://www.amazon.com</a></td>
<td>Amazon.com – Earth’s Biggest Selection</td>
</tr>
<tr>
<td>12</td>
<td><a href="http://www.newline.com">http://www.newline.com</a></td>
<td>New Line Cinema</td>
</tr>
<tr>
<td>13</td>
<td><a href="http://www.mrqe.com">http://www.mrqe.com</a></td>
<td>Movie Review Query Engine</td>
</tr>
<tr>
<td>14</td>
<td><a href="http://www.mrqe.com">http://www.mrqe.com</a></td>
<td>Movie Review Query Engine</td>
</tr>
<tr>
<td>15</td>
<td><a href="http://www.allmovie.com">http://www.allmovie.com</a></td>
<td>The Internet Movie Database (IMDb)</td>
</tr>
</tbody>
</table>

Table B.24: Top 15 results for query “movies”
<table>
<thead>
<tr>
<th>Page</th>
<th>Full Title</th>
<th>URL</th>
<th>Hits</th>
<th>Norm</th>
<th>HITS</th>
<th>HoIAvg</th>
<th>AT-Avg</th>
<th>Max</th>
<th>SALEA</th>
<th>SelRk</th>
</tr>
</thead>
</table>

**Table B.25: Top 15 results for query “national parks”**
Table B.27: Top 15 results for query “randomized algorithms”
**Table B.30: Top 15 results for query “search engines”**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
<th>HubAvg</th>
<th>SALSA</th>
<th>AT-Avg</th>
<th>Norm (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AltaVista</td>
<td><a href="http://www.altavista.com">http://www.altavista.com</a></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Google</td>
<td><a href="http://www.google.com">http://www.google.com</a></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Yahoo!</td>
<td><a href="http://www.yahoo.com">http://www.yahoo.com</a></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Ask Jeeves</td>
<td><a href="http://www.askjeeves.com">http://www.askjeeves.com</a></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Lycos</td>
<td><a href="http://www.lycos.com">http://www.lycos.com</a></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>AltaVista Text-Only Search</td>
<td><a href="http://www.raggingsearch.altavista.com">http://www.raggingsearch.altavista.com</a></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Excite France</td>
<td><a href="http://www.excite.fr">http://www.excite.fr</a></td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>My Excite</td>
<td><a href="http://www.mymexcite.com">http://www.mymexcite.com</a></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Northern Light</td>
<td><a href="http://www.northernlight.com">http://www.northernlight.com</a></td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>Dogpile</td>
<td><a href="http://www.dogpile.com">http://www.dogpile.com</a></td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>Euroseek</td>
<td><a href="http://www.euroseek.net">http://www.euroseek.net</a></td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>Ask Jeeves</td>
<td><a href="http://www.askjeeves.com">http://www.askjeeves.com</a></td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>LookSmart</td>
<td><a href="http://www.looksmart.com">http://www.looksmart.com</a></td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Ask Jeeves - THE POWERFUL SEARCHENGINE!</td>
<td><a href="http://www.askjeeves.co.uk">http://www.askjeeves.co.uk</a></td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>Empty title field</td>
<td></td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Rank</td>
<td>Query</td>
<td>URL 1</td>
<td>URL 2</td>
<td>URL 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------------------</td>
<td>--------------------------------------------</td>
<td>--------------------------------------------</td>
<td>--------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Mr. William Shakespeare and the Internet</td>
<td><a href="http://www.shakespeare.com">http://www.shakespeare.com</a></td>
<td><a href="http://www.rdg.ac.uk/globe">http://www.rdg.ac.uk/globe</a></td>
<td><a href="http://the-tech.mit.edu/Shakespeare">http://the-tech.mit.edu/Shakespeare</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.31: Top 15 results for query “shakespeare”
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Awesome Dodge Chargers for sale immediately! Mopar muscle!</td>
<td><a href="http://adlistings.vintagecars.about.com">http://adlistings.vintagecars.about.com</a></td>
</tr>
<tr>
<td>2</td>
<td>VINTAGE POSTCARD &amp; EPHEMERA</td>
<td><a href="http://vintagepostcard1.tripod.com">http://vintagepostcard1.tripod.com</a></td>
</tr>
<tr>
<td>3</td>
<td>Home,die-cast-models,vintage die cast,die cast merchandise,vinyl</td>
<td><a href="http://www.vintagediscast.com">http://www.vintagediscast.com</a></td>
</tr>
<tr>
<td>4</td>
<td>Home,die-cast-models,vintage die cast,die cast merchandise,vinyl</td>
<td><a href="http://www.vintagediscast.com">http://www.vintagediscast.com</a></td>
</tr>
<tr>
<td>5</td>
<td>2208,Welcome to Home Die Cast,Muslin,Electronic,CNC,Diecast,Buddy</td>
<td><a href="http://www.barnhillminnesota.com">http://www.barnhillminnesota.com</a></td>
</tr>
<tr>
<td>7</td>
<td>Vintage Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>9</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>10</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>11</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>12</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>13</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>14</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
<tr>
<td>15</td>
<td>Classic Cars - Classic Cars</td>
<td><a href="http://www.classiccar.com">http://www.classiccar.com</a></td>
</tr>
</tbody>
</table>

Table B.33: Top 15 results for query “vintage cars”
Table B.34: Top 15 results for query “weather”