# PageRank アルゴリズム
# およびそれに関連する研究について

京都大学 総合人間学部
認知情報学系 数理情報論分野
宮嶋健人

2009 年 1 月 30 日

## 論文内容の要旨

認知情報学系　宮嶋健人

本稿は、検索エンジン Google の PageRank とそれに関連する研究を紹介し、PageRank についての正確な理解を行うことを目的とする。

PageRank とは、全世界の Web ページのハイパーリンクの構造を利用することで Web ページの相対的な重要度を決定し、Web ページの順位付けを行うアルゴリズムである。PageRank 値は、Web グラフ上をランダムに遷移するランダムサーファーが遷移を無限回繰り返した後に、そのランダムサーファーが各 Web ページへどれだけ遷移したかを示す存在確率として考えることができる。PageRank 値を算出することは、ランダムサーファーの遷移確率行列の最大固有値 1 に対応する左固有ベクトルを算出することに他ならない。

本稿の目的を達成するために、1 節では PageRank の誕生の背景として、1998 年時点の商用検索エンジンの検索結果の質とスパムとの関係について述べる。

2 節では、PageRank に関連する基本的な事項について述べる。まず本稿における検索エンジンを定義し、Google のクロール部・インデックス部・Web サーバ部のしくみについて紹介するとともに、PageRank 値の計算に必要なマルコフ連鎖などの数学的事項や、リンク解析について紹介する。

3 節では、1998 年に発表された Google および PageRank についての原著論文 [48][8] を基に、PageRank 値の直感的な考え方、定義、計算方法について述べるとともに、PageRank の研究者達による原著論文の解釈の違いについて紹介する。

4 節では、PageRank に関連する改良研究のテーマを「PageRank 値の改良による検索結果の質の向上」「PageRank 値の計算にかかる記憶領域の節約」「PageRank 値の計算の高速化」「PageRank 値の更新の高速化」に分類して紹介する。

5 節では、本稿のまとめを行う。

# 目次

1 **はじめに** 1

2 **基本的事項** 3
2.1 検索エンジンの定義 . . . 3
2.2 検索エンジン Google . . . 4
2.2.1 クロール部 . . . 5
2.2.2 インデックス部 . . . 6
2.2.3 Web サーバ部 . . . 8
2.3 マルコフ連鎖 . . . 9
2.3.1 ノルム . . . 10
2.3.2 確率行列 . . . 11
2.3.3 マルコフ行列 . . . 12
2.3.4 エルゴード的マルコフ連鎖 . . . 13
2.3.5 定常分布 . . . 13
2.3.6 Perron-Frobenius の定理 . . . 14
2.3.7 Primitivity . . . 15
2.3.8 マルコフ行列のべき乗の収束条件 . . . 16
2.3.9 べき乗法 . . . 16
2.4 リンク解析 . . . 17
2.4.1 HITS . . . 18

3 PageRank 20
3.1 PageRank 値の直感的な考え方 . . . 20
3.2 ランダムサーファーモデル . . . 21
3.3 Simplified PageRank . . . 23
3.4 Normal PageRank . . . 27
3.5 PageRank 値の定義の訂正 . . . 28
3.6 べき乗法による PageRank 値の計算 . . . 29
3.7 Meyer らによる PageRank 値の定義 . . . 31

4 PageRank の改良 35
4.1 PageRank 値の改良による検索結果の質の向上 . . . . . 36
4.1.1 パーソナライズドランクソース . . . . . 36
4.1.2 知的サーファーモデル . . . . . 37
4.1.3 damping factor によるリンクスパムの特定 . . . . . 40
4.1.4 バックボタンモデル . . . . . 41
4.2 連立一次方程式による PageRank 値の計算 . . . . . 46
4.3 PageRank 値の計算にかかる記憶領域の節約 . . . . . 49
4.3.1 隣接リストによるハイパーリンク行列の圧縮 . . . . . 50
4.4 PageRank 値の計算の高速化 . . . . . 53
4.4.1 ダングリングノードの状態集約 . . . . . 53
4.4.2 収束判定基準の変更による効率化 . . . . . 55
4.4.3 Extrapolation . . . . . 56
4.4.4 BlockRank . . . . . 57
4.5 PageRank 値の更新の高速化 . . . . . 59

5 おわりに 61

謝辞 63

参考文献 63

# 1 はじめに

本稿は検索エンジン Google の PageRank とそれに関連する研究について紹介し、検索結果の質を高めることに成功した PageRank について正しい理解を行うことを目的とする。

PageRank とは、Sergey Brin と Lawrence Page が発表した、Web のハイパーリンク構造を利用して Web ページを順位付けするアルゴリズムである [48]。1998 年 9 月、Brin らは PageRank[*1]を採用した Web 検索エンジン：Google[*2]を公開した。Google は検索結果の質を高めることを目標の一つとしている [8]。

Google の誕生の背景として、Brin らは当時の検索エンジンの検索結果の質の悪さについて述べている [8]。Brin らは、1997 年の 11 月に主要な 4 社の商用検索エンジンを使ってそれらの検索エンジン自身を検索したが、それらのうち 1 社の検索エンジンしか、自分自身を最初の 10 件の検索結果内に表示することができなかった。

旧郵政省の統計 [65] からも、当時の検索エンジンの検索結果の質の悪さを推測することができる。1996 年の日本のインターネットユーザの 29.2% が「インターネットの需要は今後伸びない」と答えている。その理由としては「思ったほど有益な情報がない」(43.8%) が最も多かった。当時の検索エンジンは Web ページ内の単語情報のみで Web ページの重要度を決定しており、有益な情報をもたない Web ページでもワードスパム[*3]を行えば検索結果の上位に表示されることがあった [46]。ワードスパムによる検索結果の質の低下が、ユーザに「思ったほど有益な情報がない」と感じさせた原因の一つとなった可能性がある。

Google は検索結果の質を高めるために、単語情報に加えてアンカーテキスト[*4]および Web ページ間のリンク構造を解析するアルゴリズム：PageRank を利用して Web ページの重要度を決定している [8]。Google はある Web ページからある Web ページに対するリンクを支持投票とみなし、重要な Web ページからリンクされている Web ページは重要な Web ページであるとして、それには高い PageRank を与える [19]。

PageRank を意図的かつ不当に高めようとするリンクスパム[*5]も存在するが、リンクス

[*1] "PageRank" という名前は、"Lawrence Page" と "Web Page" をかけたものであると考えられている。
[*2] http://www.google.com/
[*3] 隠しテキストや詰め込みテキストなどの手法がある [22]。
[*4] Anchor Text。リンクにつけられた文字列のこと。HTML 中で <A> タグで囲まれている。
[*5] リンクファーム、画像による隠しリンク、リンクエクスチェンジなどの手法がある。

パムを特定する研究も行われている [55][15]。また、Google は PageRank の不正な操作を意図した Web ページにペナルティを与えている [21]。ACSI[*6]によると、米国では主要な 6 社の検索エンジンのうち Google のユーザ満足度が 2002 年から 2008 年現在まで 1 位 (2007 年のみ 2 位) を記録している [30]。ユーザが Google を支持する原因の一つとして、スパムによる検索結果の質の低下を防いでいることが挙げられる。

本稿の構成は以下のとおりである。2 節では、PageRank に関連する基本的な事項について述べる。まず本稿における検索エンジンを定義し、検索エンジン Google のクロール部・インデックス部・Web サーバ部のしくみについて紹介するとともに、PageRank 値の計算に必要なマルコフ連鎖などの数学的事項や、リンク解析について紹介する。3 節では、1998 年に発表された Google および PageRank についての原著論文 [48][8] を基に、PageRank 値の直感的な考え方、定義、計算方法について述べるとともに、PageRank の研究者による原著論文の解釈の違いについて紹介する。4 節では、PageRank に関連する改良研究のテーマを「PageRank 値の改良による検索結果の質の向上」「PageRank 値の計算にかかる記憶領域の節約」「PageRank 値の計算の高速化」「PageRank 値の更新の高速化」に分類して紹介する。5 節では、本稿のまとめを行う。

[*6] American Customer Satisfaction Index

# 2 基本的事項

本節では、基本的事項として検索エンジンを定義し、検索エンジン Google 全体のしくみを紹介する。また、次節以降で扱う PageRank 値の計算に必要な知識としてマルコフ連鎖を紹介するとともに、PageRank 以外のハイパーリンクの解析モデルの代表例として HITS を紹介する。

## 2.1 検索エンジンの定義

本小節では、検索エンジンという言葉の意味を定義する。

一般に、検索機能を提供するシステムのことを検索エンジンとよぶ。検索エンジンは Web だけでなく図書館の蔵書検索などでも使われている。Web で公開されているリソース（Web ページ、画像ファイルなど）の位置を検索する Web 検索エンジンのことを単に検索エンジンとよぶこともある [41]。

検索エンジンを提供している Web サイト全体を指して検索エンジンとよぶこともある [67] が、本稿では、このような Web サイトは検索サイトまたはポータルサイト (portal site) とよぶ。

[61] では、検索エンジンを文書のデータベースの作成方法の違いによりディレクトリ型検索エンジンとロボット型検索エンジンとに大別している。ディレクトリ型検索エンジンでは、人間が Web ページをディレクトリ（カテゴリ）に分類してデータベースを作成する[*7]。また、ロボット型検索エンジンではロボット（クローラ、スパイダ）と呼ばれるソフトウェアを使い、Web コンテンツすなわち Web のハイパーテキストや PDF、画像などのファイルを自動収集してデータベースを作成する[*8]。Google はロボット型検索エンジンの代表例である。ロボット型検索エンジンは、収集した Web コンテンツに順位付けをおこない、ユーザの入力クエリ[*9]に適合した文書を、ランクの高い順番に並べて検索結果を表示する。

[67] では、ロボット型検索エンジンをキーワード検索型検索エンジンと全文検索型検索エンジンに大別している。キーワード検索型検索エンジンではデータベースに文書と関連

[*7] 例：Yahoo! Directory (http://dir.yahoo.com/)

[*8] 例：Google(http://www.google.com/)、Yahoo!(http://www.yahoo.com/)、MSN（http://www.msn.com/)、百度（http://www.baidu.com/）など。

[*9] ユーザが検索時に入力するキーワードの組み合わせのこと。また、入力クエリを使って検索結果を要求する操作をクエリ入力という。

するキーワードを登録する。キーワードの登録は人の手で行うか、文書の一部を自動抽出して行う。ユーザの入力クエリと登録キーワードとが合致した場合、キーワードの登録元の文書を適合した文書として検索結果に表示する。それに対し、全文検索型検索エンジンでは文書内の全文を検索対象にする。ユーザの入力クエリと文書内の語句に合致するものがあれば、その語句が含まれる文書を適合した文書として検索結果に表示する。[67] では、ポータルサイトの多くがキーワード検索型ではなく全文検索型を採用していると述べている[*10]。

[67] では、全文検索型検索エンジンを逐次検索型とインデックス（索引）検索型とに大別している。逐次検索型では、ユーザの入力キーワードとデータベース内の全ての文書の全文とを照合し、合致した文書を検索結果に表示させる。神崎らは、この方法は文書の数が多いほど時間がかかるため Web 上の膨大な文書を扱うのには向いていないと述べ、検索エンジンの多くはインデックス検索を採用していると述べている [67]。インデックス検索では、あらかじめ検索対象の文書を走査して索引情報を作成しておき、ユーザのクエリ入力に応じて索引情報を参照して、合致した文書を検索結果に表示する。

以下、本稿で単に検索エンジンといった場合には、インデックス検索を行う全文検索型のロボット型 Web 検索エンジンシステムのことを指すものとする。

## 2.2 検索エンジン Google

前小節では本稿での検索エンジンという言葉の意味を定義した。本小節では、検索エンジン Google のしくみについて説明する。

一般に、検索エンジンのしくみは 3 つの機構に分けて説明されることが多い。[41] では、クローラ（またはスパイダ）、インデックス、インターフェースの 3 つから成ると述べている。また、[61] では、検索バックエンド、インデックス、検索サーバの 3 つから成ると述べている。本稿では、Brin らが [8] で用いたクローラ、インデックス、Web サーバという呼称を用いる。

一般に、クローラ部では全世界の Web ページを訪れて情報を収集する。また、インデックス部ではクローラが集めた情報をデータベースに記憶し、すばやく検索結果を出力できるよう前処理を行う。Web サーバ部ではユーザのクエリ入力を受け、入力クエリに適合した文書に順位付けを行い検索結果を HTML 形式で出力する。

これら 3 つの機構の性能が、検索エンジンの検索結果の質やクエリ応答時間[*11]を決定

[*10] もしくは、全文検索型とあわせてキーワード検索型の検索結果を表示している [67]。
[*11] ユーザのクエリ入力を受けてから、ユーザに検索結果を表示するまでに要した時間。

する要因の一つになる [67]。例えば、クローラが多くの情報を集めるほど、インデックスのもつ情報量は大きくなり、検索結果のヒット数[*12]は大きくなる [41]。Web 上の情報を速く集められるようなクロール戦略を立て、収集した情報をインデックス部で素早く記憶・処理できれば、最新の Web サイトの情報を検索結果に反映することができるであろう。

Google は効率的なクロール、インデックス情報の効率的な読み書き、迅速な検索結果の表示を目標の一つとしている [8]。特に、クロールの頻度とインデックス情報の更新は PageRank の更新および検索結果の鮮度に大きくかかわるであろう。以下で、1998 年の Brin らの論文 [8] をもとに、Google のクロール部・インデックス部・Web サーバ部について説明する。

### 2.2.1 クロール部

クロール（またはクローリング: crawling）とは、Web ページを訪れて情報を収集することをいう [61]。クロールを行うソフトウェアをクローラ（crawler）とよぶ[*13]。

Brin らは、1998 年時点の Google はクローラを複数のマシンに分散させて同時に動かしてクロールをおこなっていると述べている [8]。クローラの管理は URL サーバが行い、通常は 1 つの URL サーバが 3 つのクローラに対して巡回する URL のリストを渡す。各クローラは巡回する URL のリストをもとに、それぞれ同時に 300 個程度の Web サーバへアクセスし、それらの Web ページの情報をダウンロードして記憶サーバに送信する。記憶サーバは各 Web ページに「docID」(各 Web ページに固有な自然数値) を割り当て、リポジトリ（repository）に記憶する。リポジトリには、「docID」、「url」（Web ページの URL)、「text」（Web ページを圧縮したもの)、ダウンロードした日時や Web ページの長さなどが記憶される。クローラは URL のリストを巡回し終えたら、URL サーバから新たなリストを受け取ってクロールを再開する。クロールはこの作業の繰り返しである。

Brin らは、クロールの際に時間がかかるのは、DNS[*14]によるアドレス解決であると述べている [8]。そのためクローラは DNS キャッシュをもつことで効率的なアドレス解決を行っている。

---

[*12] ユーザの入力クエリに適合した文書の数。

[*13] またはスパイダ（spider)、ロボット（robot）とよばれる [67]。

[*14] Domain Name System の略。

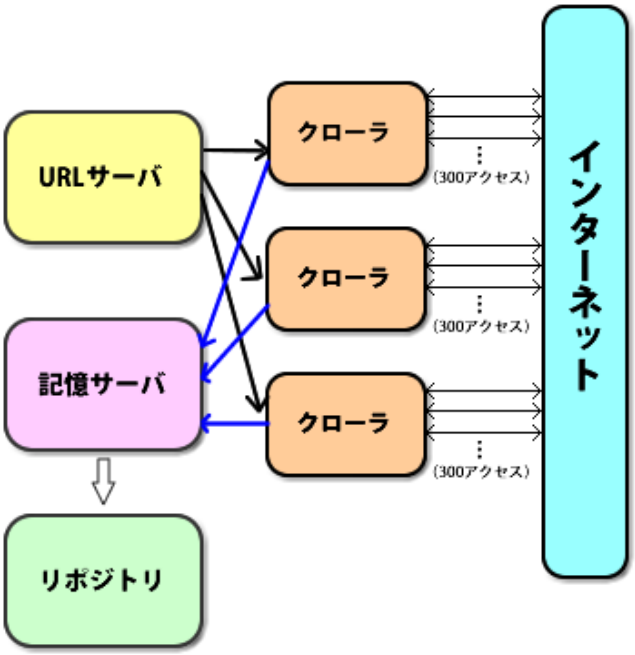


図 1　Google のクロール部の処理の流れ

### 2.2.2　インデックス部

Brin らは、Google のインデックス部の処理について次のように述べている [8]。

まず、インデクサ (Indexer) がリポジトリに記憶された各 Web ページの情報を読み込んで構文解析を行う。構文解析で得られた各 Web ページの情報 (docID、URL、タイトル) は、ドキュメントインデックス（DocIndex）に記憶される。ドキュメントインデックスは、docID をキーとし、Web ページの URL やタイトルを値として持つインデックスである。もしまだクロールされていない URL があれば、URL サーバの巡回リストに追加され、クロールが行われる。

次に、構文解析で得た各々の単語に一意的な wordID をつけて辞書（lexicon）を作成する。また、構文解析で得た単語情報[*15]をバレル（barrels）に記憶する。バレルでは、docID をキーとしてその Web ページに含まれる単語の wordID とその単語の出現位置・フォントサイズ・文字装飾の情報を値として持つ正引きインデックス（forward index）を

[*15] Web ページ内にある単語、文書内での単語の位置、フォントサイズ、文字装飾の情報。

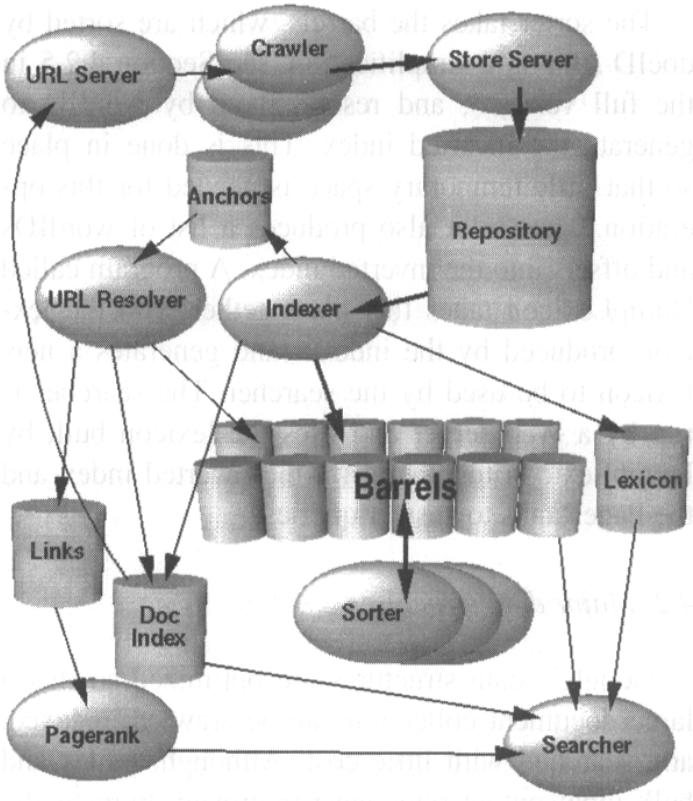


図 2　Google のインデックス部の処理の流れ　（[8] より引用）

作成する。

また、インデクサは構文解析で得たリンク情報をもとにリンク解析を行う。まず、各 Web ページ内のすべてのリンク情報[*16]をアンカー（Anchors）に記憶する。次に、URL リゾルバ（URL resolver）がアンカーに記憶されているリンク情報を読み込む。

URL リゾルバはドキュメントインデックス、バレル、リンク（Links）へ情報を送信する。ドキュメントインデックスへは、アンカーから読みこんだ相対 URL を絶対 URL に変換[*17]した情報を送信する。バレルへは各 Web ページの docID と被リンクのアンカーテキストを送信し、正引きインデックスにアンカーテキストの情報を追加する。リンク（Links）へは、どの docID からどの docID へのリンクが存在するかを送信する。リンク（Links）に登録されたリンク情報は PageRank を算出するために利用され、PageRank の

[*16] リンク元 Web ページ、リンク先 Web ページ、アンカーテキストの情報。

[*17] Web ページ内で使われている相対リンク（例：index.html）を絶対リンク（例：http://www.example.com/index.html）に変換する。

値はリンク（Links）へ記憶される。

ソータ（Sorter）は、docID をキーとしている正引きインデックスを wordID でソートして転置インデックス（Inverted index）を作成する。転置インデックスとは、wordID をキーとし、その単語が出現する Web ページの docID と各単語の出現位置・フォントサイズ・文字装飾、および docID の被リンクのアンカーテキストの情報をもつインデックスである。

### 2.2.3　Web サーバ部

Brin らは、Web サーバについて以下のように述べている [8]。

Web サーバは、ユーザの入力クエリに適合した Web ページを重要度の高い順番に並び替えて検索結果を表示する。重要度の決定には、単語情報、アンカーテキスト、PageRank

表 1　Google のアーキテクチャの処理システムと記憶システム

| 処理システム | 役割 |
|---|---|
| URL サーバ | クローラに対して、巡回する URL を指示。 |
| クローラ | Web ページの情報をダウンロードし、記憶サーバに送信。 |
| 記憶サーバ | docID を割り当て、Web ページの情報をリポジトリに送信。 |
| インデクサ | 構文解析・辞書作成・リンク解析を行う。 |
| URL リゾルバ | リンク情報をドキュメントインデックス、バレル、リンクに送信。 |
| ソーター | 転置インデックスを作成する。 |

| 記憶システム | |
|---|---|
| リポジトリ | docID をキーとし、URL・HTML などを記憶。 |
| アンカー | docID をキーとし、リンク元・リンク先・アンカーテキストを記憶。 |
| バレル | 正引きインデックス、転置インデックスを記憶。 |
| 正引きインデックス | docID をキーとし、wordID・単語情報・アンカーテキストを記憶。 |
| 転置インデックス | wordID をキーとし、docID・単語情報・アンカーテキストを記憶。 |
| 辞書 | 単語をキーとし、wordID を値とするインデックスを記憶。 |
| ドキュメントインデックス | docID をキーとし、Web ページの URL・タイトルを記憶。 |
| リンク（Links） | docID をキーとし、PageRank 値を値とするインデックスを記憶。 |

の 3 つの要素を用いる。

Brin らは、Web サーバの動作について以下のように述べている [8]。まず、ユーザの入力クエリを構文解析して単語に分解する。そして辞書を参照して各単語の wordID を調べる。入力クエリの wordID をキーとして転置インデックスを引き、すべての wordID に適合したすべての Web ページの docID を並べたリストを作成する。

適合した Web ページをランク順に並び替えるために、正引きインデックスを参照して各 wordID に適合した docID の単語情報・アンカーテキストを調べるとともに、リンク（Links）を参照し PageRank を調べる。3 つの要素のスコアを総合して docID の順位付けを行い、ランキング上位の docID のドキュメントインデックスを参照する。ドキュメントインデックスから Web ページのタイトルや URL の情報を読み込み、検索結果の上位一部（10 件、20 件など）を HTML の形式で送信する。

## 2.3 マルコフ連鎖

この節では、PageRank アルゴリズムの理解に必要となるマルコフ連鎖（Markov chain）、ノルム、確率行列、マルコフ行列とその定常分布、べき乗法について、それらの定義と性質を紹介する。以下で必要な語句の定義を述べていく。詳しくは [42][58][60] を参照するとよい。

Page らは、PageRank アルゴリズムの直感的な説明にランダムサーファーモデル (3.2 節で後述）を用いている。大まかにいうと、ランダムサーファーモデルとはランダムにハイパーリンクを辿っていく多数の Web サーファー（ランダムサーファー）が Web ページの遷移を無限回繰り返して定常状態に達したときに、ある Web ページを閲覧している割合をその Web ページの相対的な重要度とみなす、というモデルである [48]。

ランダムサーファーの遷移は現在閲覧している Web ページにのみ依存する。Meyer らは [35] において、Page らが PageRank の説明を行うにあたってマルコフ連鎖という言葉を使わずにランダムサーファーモデルのみを用いていることを指摘している。その上で Meyer らは、ランダムサーファーモデルには既存のマルコフ連鎖についての研究結果により適切な解が存在することが保証されると述べている [42][35]。

有限状態空間 $S = \{S_1, S_2, \ldots, S_n\}$ とし、時点を表すパラメータを $t(t = 0, 1, 2, \ldots)$ とする。$S$ を状態空間とする確率変数の列 $\{X_t\}_{t=0}^{\infty}$ を確率過程[*18]とよぶ。

[*18] 一般に、状態空間の状態数は有限の場合と無限の場合とが考えられる。また、時点 $t$ は連続的である場合と離散的である場合とが考えられるが、ここでは状態数有限かつ離散時間である確率過程のことを、単に確率過程とよぶこととする。

確率過程 $\{X_t\}_{t=0}^{\infty}$ がマルコフ性を持つとは、次が成り立つ場合をいう。

$$P(X_{t+1} = S_j \mid X_t = S_{i_t}\ ,\ X_{t-1} = S_{i_{t-1}}\ ,\ \ldots\ ,\ X_0 = S_{i_0}) = P(X_{t+1} = S_j \mid X_t = S_{i_t}) \quad (2.1)$$

左辺は、時点 0 で状態 $S_{i_0}$、...、時点 $t$ で状態 $S_{i_t}$ であるときに時点 $t+1$ で状態 $S_j$ である確率である。右辺は、時点 $t$ で状態 $S_{i_t}$ であるときに時点 $t+1$ で状態 $S_j$ である確率である。マルコフ性をもつ確率過程をマルコフ連鎖とよぶ。上式が意味するのは、マルコフ連鎖では、時点 $t+1$ において状態 $S_j$ にいる確率は、時点 $t$ における状態 $S_{i_t}$ にのみ依存するということである。

### 2.3.1 ノルム

以下で、マルコフ連鎖について解析するときに必要となるベクトルのノルムおよび作用素ノルムの定義を紹介する [35]。

ベクトル $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ または $\mathbb{C}^n$[*19]について、以下の条件を満たす関数

$$||\cdot|| : \mathbb{R}^n \text{または} \mathbb{C}^n \to \mathbb{R}$$

をノルム（ベクトルノルム）とよぶ。ベクトル $\mathbf{x} = {}^t(x_1, x_2, \ldots, x_n)$ とする。

- $||\mathbf{x}|| \geq 0$　　（等号が成り立つのは $\mathbf{x} = 0$ のときのみ。）
- $||\alpha\mathbf{x}|| = |\alpha|||\mathbf{x}||$　　（$\alpha$ : 実数または複素数）
- $||\mathbf{x} + \mathbf{y}|| = ||\mathbf{x}|| + ||\mathbf{y}||$

式 (2.2) で 1-ノルム、式 (2.3) で無限大ノルムをそれぞれ定義する。

$$||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i| \quad (2.2)$$

$$||\mathbf{x}||_\infty = \max_i |x_i| \quad (2.3)$$

一般に、以下の式で定義されるノルムを $p$-ノルムとよぶ $(1 \leq p)$。

$$||\mathbf{x}||_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \quad (2.4)$$

ベクトルのノルムは横ベクトルに対しても同様に定義しておくものとする。

[*19] 本稿に出てくるベクトルは全て数値横ベクトルか数値縦ベクトルである。

行列 $\mathbf{A}_{m\times n}$ を実あるいは複素行列とする。ベクトル空間 $U^m$, $V^n$ にそれぞれノルム $||\cdot||_{U^m}$ , $||\cdot||_{V^n}$ が定められているとき、それらのノルムに対応する行列 $\mathbf{A}$ の作用素ノルム[*20]$||\mathbf{A}||$ を以下の式で定義する。

$$||\mathbf{A}|| = \max_{0\neq\mathbf{x}\in V^n} \frac{||\mathbf{A}\mathbf{x}||_{U^m}}{||\mathbf{x}||_{V^n}} = \max_{||\mathbf{x}||_{V^n}=1} ||\mathbf{A}\mathbf{x}||_{U^m} \tag{2.5}$$

するとこれは $m\times n$ 行列全体をベクトル空間としてみたときにノルムの条件を満たす。$m=n$ かつ $||\cdot||_{U^m} = ||\cdot||_{V^n} = ||\cdot||_\star$ のとき、$||\mathbf{A}||$ を $||\mathbf{A}||_\star$ と書くことにする。

行列 $\mathbf{A}_{n\times n}$ を実あるいは複素行列とする。1-ノルムにより導かれる作用素ノルム、および無限大ノルムにより導かれる作用素ノルムをそれぞれ以下のように定義する。

$$||\mathbf{A}||_1 = \max_{||\mathbf{x}||_1=1} ||\mathbf{A}\mathbf{x}||_1 = \max_j \sum_i |a_{ij}| \tag{2.6}$$

$$||\mathbf{A}||_\infty = \max_{||\mathbf{x}||_\infty=1} ||\mathbf{A}\mathbf{x}||_\infty = \max_i \sum_j |a_{ij}| \tag{2.7}$$

本稿では、行列 $\mathbf{A}_{n\times n}$ の固有値のうち、固有値の絶対値の最大値をスペクトル半径とよび、$\rho(\mathbf{A})$ と表す。

$$\rho(\mathbf{A}) = \max_{\lambda\in\sigma(\mathbf{A})} |\lambda|$$

ただし $\sigma(\mathbf{A})$ は行列 $\mathbf{A}$ のすべての固有値を元とする集合である[*21]。

任意のベクトルノルム $||\cdot||_\star$ に対し、行列 $\mathbf{A}$ のスペクトル半径は、$||\cdot||_\star$ に対応する行列 $\mathbf{A}$ の作用素ノルム $||\mathbf{A}||_\star$ を超えない [64]。

$$\rho(\mathbf{A}) \leq ||\mathbf{A}||_\star \tag{2.8}$$

### 2.3.2 確率行列

本稿では、実ベクトル $\mathbf{x}$ について $\mathbf{x}$ のすべての成分が非負である場合、$\mathbf{x}$ は非負である（非負ベクトルである）といい $\mathbf{x}\geq 0$ と書くことにする。またすべての成分が正である場合、$\mathbf{x}$ は正である (正ベクトルである）といい $\mathbf{x}>0$ と書くことにする。

同様に、実行列 $\mathbf{A}$ について行列 $\mathbf{A}$ のすべての成分が非負である場合、行列 $\mathbf{A}$ は非負である（非負行列である）といい $\mathbf{A}\geq 0$ と書くことにする。またすべての成分が正である場合、行列 $\mathbf{A}$ は正である（正行列である）といい $\mathbf{A}>0$ と書くことにする。

---

[*20] operator norm。自然ノルム（natural norm）または誘導ノルム (induced norm) ともよぶ。

[*21] $\sigma(\mathbf{A})$ を行列 $\mathbf{A}$ のスペクトルとよぶ。

各行の成分の合計が 1 となるような非負行列 $\mathbf{P}_{n\times n}$ $(n \geq 1)$ を確率行列 (stochastic matrix) とよぶ[*22]。

確率行列 $\mathbf{P}_{n\times n}$ のスペクトル半径について、$\rho(\mathbf{P}) = 1$ である [35]。行列 $\mathbf{P}$ は各行の成分の合計が 1 であるから、$\mathbf{1} = {}^t(1\ \ldots\ 1)$ として、この条件を

$$\mathbf{P1} = \mathbf{1} \tag{2.9}$$

と書ける。このとき、$||\mathbf{1}||_\infty = 1$ と式 (2.7) から $||\mathbf{P}||_\infty = 1$ である。また、式 (2.9) より確率行列は固有値に 1 をもつ。よって式 (2.8) より、

$$1 \leq \rho(\mathbf{P}) \leq ||\mathbf{P}||_\infty = 1 \tag{2.10}$$

である。従って、確率行列 $\mathbf{P}$ のスペクトル半径は

$$\rho(\mathbf{P}) = 1 \tag{2.11}$$

である。

### 2.3.3 マルコフ行列

式 (2.1) のマルコフ連鎖は時点 $t-1$ から時点 $t$ へ遷移する際の条件付き確率によって規定される。時点 $t-1$ において状態 $S_i$ へ遷移しており、かつ時点 $t$ において状態 $S_j$ へ遷移している確率を遷移確率とよび、

$$P(X_t = S_j | X_{t-1} = S_i) = p_{ij} \tag{2.12}$$

と表す。遷移確率が時点によって変化しない ($p_{ij}(t) = p_{ij}$) マルコフ連鎖のことを定常マルコフ連鎖ともよぶ。本稿で単にマルコフ連鎖といった場合には、定常マルコフ連鎖のことを指すものとする。また、遷移確率 $p_{ij}$ を行列で表現したものを遷移確率行列とよび、

$$\mathbf{P}_{n\times n} = [p_{ij}] \tag{2.13}$$

である。遷移確率行列 $\mathbf{P}$ をマルコフ行列ともよぶ。マルコフ連鎖のマルコフ行列 $\mathbf{P}$ は、各行の成分の合計が 1 となる非負行列であるから、確率行列である。また逆に確率行列はあるマルコフ連鎖のマルコフ行列になっている。

[*22] 各列の合計が 1 である場合にも確率行列と呼ぶことがある。各行の合計が 1 である確率行列を明示的に”row-stochastic”な確率行列とよぶ場合がある。

### 2.3.4 エルゴード的マルコフ連鎖

$n$ 次正方行列 $\mathbf{A} = [a_{ij}]$ が可約 (reducible) であるとは、$\mathbf{A}$ の有向グラフ[*23]が強連結[*24]でない場合をいう。それに対して、$n$ 次正方行列 $\mathbf{A}$ が既約 (irreducible) であるとは、$\mathbf{A}$ の有向グラフが強連結である場合をいう。あるマルコフ連鎖のマルコフ行列が可約であれば、そのマルコフ連鎖を可約なマルコフ連鎖とよぶ。同様に、あるマルコフ連鎖のマルコフ行列が既約であれば、そのマルコフ連鎖を既約なマルコフ連鎖とよぶ。

状態空間の全部または一部が $r$ 個の部分集合 $D_1,\ D_2,\ \cdots,\ D_r$ に分割されているとする。$D_1$ の中の状態から遷移を開始した場合、

$$D_1 \to D_2 \to \cdots \to D_r \to D_1 \to \cdots \tag{2.14}$$

の順番に各部分集合の中の状態へ遷移し、他の遷移の可能性がないような既約なマルコフ連鎖があるとする。このときの整数 $r$ のうち最大のものをマルコフ連鎖の周期とよび、$r \geq 2$ である既約なマルコフ連鎖を周期的なマルコフ連鎖とよぶ。また、$r = 1$ である既約なマルコフ連鎖をエルゴード的マルコフ連鎖（非周期的なマルコフ連鎖）とよぶ [60]。

### 2.3.5 定常分布

非負の行ベクトル ${}^t\mathbf{p} = (p_1, p_2, \ldots, p_n)$ で $\sum_k p_k = 1$ となるようなものを確率分布ベクトルという。状態数が $n$ 個のマルコフ連鎖の $k$ 回目の遷移時点の確率分布ベクトル ${}^t\mathbf{p}(k)$ を次の式で定義する。

$${}^t\mathbf{p}(k) = (p_1(k), p_2(k), \ldots, p_n(k)) \quad \text{ただし} \quad p_j(k) = P(X_k = S_j)$$

特に、${}^t\mathbf{p}(0)$ を初期分布ベクトル（初期分布）とよぶ。

$${}^t\mathbf{p}(0) = (p_1(0), p_2(0), \ldots, p_n(0)) \quad \text{ただし} \quad p_j(0) = P(X_0 = S_j)$$

$p_j(0)$ は、マルコフ連鎖の遷移の開始時点での状態が $S_j$ である確率を表している。

初期ベクトル ${}^t\mathbf{p}(0)$ を用いて、1 回目の遷移後の確率分布ベクトル ${}^t\mathbf{p}(1)$ を次のように

---

[*23] 1,2,...,n の節点に対し、$a_{ij} \neq 0$ ならば節点 $i$ から節点 $j$ への有向辺をもち、$a_{ij} = 0$ ならば有向辺をもたないグラフのことをいう。

[*24] 任意の 2 節点について、有向辺をそれらの向きに辿って互いに到達可能である場合をいう。

求める。すべての状態 $j$ について、1 回目の遷移後に状態 $j$ にいる確率は、

$$\begin{aligned}
p_j(1) &= P(X_1 = S_j) \\
&= P(X_1 = S_j \wedge (X_0 = S_1 \vee X_0 = S_2 \vee \ldots \vee X_0 = S_n)) \\
&= P((X_1 = S_j \wedge X_0 = S_1) \vee (X_1 = S_j \wedge X_0 = S_2) \vee \ldots \vee (X_1 = S_j \wedge X_0 = S_n)) \\
&= \sum_{i=1}^{n} P(X_1 = S_j \wedge X_0 = S_i) \\
&= \sum_{i=1}^{n} P(X_0 = S_i) P(X_1 = S_j | X_0 = S_i) \\
&= \sum_{i=1}^{n} p_j(0) p_{ij}
\end{aligned}$$

である [35]。よって ${}^t\mathbf{p}(1) = {}^t\mathbf{p}(0)\mathbf{P}$ である。同様に、${}^t\mathbf{p}(2) = {}^t\mathbf{p}(1)\mathbf{P}$ , ${}^t\mathbf{p}(3) = {}^t\mathbf{p}(2)\mathbf{P}$ , ... である。すなわち $k$ 回目の遷移後の確率分布ベクトルは、${}^t\mathbf{p}(k) = {}^t\mathbf{p}(0)\mathbf{P}^k$ で求めることができる

${}^t\boldsymbol{\pi}\mathbf{P} = {}^t\boldsymbol{\pi}$ を満たすような確率分布ベクトル ${}^t\boldsymbol{\pi}$ を、マルコフ行列 $\mathbf{P}$ の定常分布 (stationary distribution)[*25]とよぶ。$\mathbf{P}^k$ が $k$ 回のべき乗で収束する場合、$\mathbf{P}^{k+1} = \mathbf{P}^k$ なので $({}^t\mathbf{p}(0)\mathbf{P}^k)\mathbf{P} = {}^t\mathbf{p}(0)\mathbf{P}^{k+1} = {}^t\mathbf{p}(0)\mathbf{P}^k$ より定常分布 ${}^t\boldsymbol{\pi}$ を ${}^t\mathbf{p}(0)\mathbf{P}^k$ として求めることができる。定常分布 ${}^t\boldsymbol{\pi}$ は、マルコフ行列 $\mathbf{P}$ の最大固有値 1 に対応する左固有ベクトルである。[*27]

エルゴード的マルコフ連鎖では、${}^t\mathbf{p}(k)$ が初期分布 ${}^t\mathbf{p}(0)$ によらず一意的な定常分布 ${}^t\boldsymbol{\pi}$ に収束する [60]。詳細な証明は [42][35] を参照すること。2.3.6 節および 2.3.7 節および 2.3.8 節において、その証明の一部を紹介する。

### 2.3.6 Perron-Frobenius の定理

実正方行列 $\mathbf{A}$ が正行列である場合、次の性質が成り立つ [42]。なお、行列 $\mathbf{A}$ の固有多項式 $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$ の根 $r$ の重複している数を *algebraic multiplicity* とよび、$\text{alg mult}_A(r)$ と表記する。

1. $\rho(\mathbf{A}) > 0$ である。(以下 $r = \rho(\mathbf{A})$ とおく。)
2. $r \in \sigma(\mathbf{A})$
3. $\text{alg mult}_A(r) = 1$

[*25] 不変分布 (invariant distribution) ともよぶ[*26]。

[*27] マルコフ行列は確率行列であるから、式 (2.9) と式 (2.10) とにより最大固有値 1 をもつ。

4. $\mathbf{Ax} = r\mathbf{x}$ を満たす右優固有ベクトル[*28]$\mathbf{x} > 0$ が存在する。
   また ${}^t\mathbf{yA} = r\,{}^t\mathbf{y}$ を満たす左優固有ベクトル ${}^t\mathbf{y} > 0$ が存在する。
5. 以下のようなベクトル $\mathbf{p}$ がただひとつ存在する。
   - $\mathbf{Ap} = r\mathbf{p}, \quad \mathbf{p} > 0, ||\mathbf{p}||_1 = 1 \quad$ かつ
   - $\mathbf{A}$ の固有値に対応する正の右固有ベクトルは $\mathbf{p}$ を正数倍したもの以外に存在しない。

   このベクトル $\mathbf{p}$ を右 Perron ベクトルという。また、以下のようなベクトル ${}^t\mathbf{q}$ がただひとつ存在する。
   - ${}^t\mathbf{qA} = r{}^t\mathbf{q}, \quad {}^t\mathbf{q} > 0, ||{}^t\mathbf{q}||_1 = 1 \quad$ かつ
   - $\mathbf{A}$ の固有値に対応する正の左固有ベクトルは ${}^t\mathbf{q}$ を正数倍したもの以外に存在しない。

   このベクトル ${}^t\mathbf{q}$ を左 Perron ベクトルという。
6. $r$ は $\mathbf{A}$ のスペクトル円[*29]上の唯一の固有値である。

### 2.3.7 Primitivity

非負かつ既約な正方行列 $\mathbf{A}$ は、以下の条件を満たす場合に *primitive* であるという [35]。

$$\lim_{k\to\infty}\left(\frac{\mathbf{A}}{r}\right)^k = \frac{\mathbf{p}{}^t\mathbf{q}}{{}^t\mathbf{qp}} > \mathbf{0} \tag{2.15}$$

ただし、$r = \rho(\mathbf{A})$ であり、$\mathbf{p}$ および ${}^t\mathbf{q}$ はそれぞれ行列 $\mathbf{A}$ の右 Perron ベクトル、左 Perron ベクトルである。

以下は、非負の正方行列 $\mathbf{A}$ が primitive であるかどうかの判定を行う十分条件である（(ii) は必要十分条件である）[35]。

(i) 行列 $\mathbf{A}$ が既約かつ少なくとも 1 つの正の対角成分をもてば、行列 $\mathbf{A}$ は primitive である。

(ii) 行列 $\mathbf{A}$ について、$\mathbf{A}^m > \mathbf{0}$ （$^\exists m > 0$）であるなら、行列 $\mathbf{A}$ は primitive である。

primitive な行列は非周期的である [35]。

---

[*28] 絶対値が最大の固有値に対応する固有ベクトルを優固有ベクトルとよぶ。

[*29] 行列 $\mathbf{A} \in \mathbb{C}^{n\times n}$ とする。複素平面において、原点を中心とする半径 $r = \rho(\mathbf{A})$ の円をスペクトル円 (spectral circle) とよぶ。

### 2.3.8 マルコフ行列のべき乗の収束条件

行列 $\mathbf{P}_{n\times n}$ を非周期的なマルコフ連鎖のマルコフ行列とする。行列 $\mathbf{P}$ が primitive であれば $\lim_{k\to\infty}\mathbf{P}^k$ が存在する。この場合、行列 $\mathbf{P}$ の左 Perron ベクトルは ${}^t\boldsymbol{\pi}$ であり、右 Perron ベクトルは $\mathbf{1}/n$ である。マルコフ行列 $\mathbf{P}$ の最大固有値は 1 であるから式 (2.15) より、

$$\lim_{k\to\infty}\left(\frac{\mathbf{P}}{r}\right)^k=\lim_{k\to\infty}\mathbf{P}^k=\frac{(\frac{\mathbf{1}}{n})^t\boldsymbol{\pi}}{{}^t\boldsymbol{\pi}(\frac{\mathbf{1}}{n})}=\frac{\mathbf{1}^t\boldsymbol{\pi}}{{}^t\boldsymbol{\pi}\mathbf{1}}=\mathbf{1}^t\boldsymbol{\pi}=\begin{pmatrix}\pi_1&\pi_2&\dots&\pi_n\\ \pi_1&\pi_2&\dots&\pi_n\\ \vdots&\vdots&&\vdots\\ \pi_1&\pi_2&\dots&\pi_n\end{pmatrix}>\mathbf{0}$$

したがって、行列 $\mathbf{P}$ が primitive なら定常分布が存在し、以下が成り立つ。

$$\lim_{k\to\infty}{}^t\mathbf{p}(k)=\lim_{k\to\infty}{}^t\mathbf{p}(0)\mathbf{P}^k={}^t\mathbf{p}(0)\mathbf{1}^t\boldsymbol{\pi}={}^t\boldsymbol{\pi}$$

なお $\sum_k p_k(0)=1$ より、${}^t\mathbf{p}(0)\mathbf{1}=1$ である。よって、定常分布 ${}^t\boldsymbol{\pi}$ は初期分布 ${}^t\mathbf{p}(0)$ の値によらない。

### 2.3.9 べき乗法

Brin らは、1998 年時点の PageRank の計算方法としてべき乗法 (Power Method) を用いている [48]。べき乗法は絶対値最大の固有値を求めるアルゴリズムである。べき乗法を用いて、一定の条件を満たすマルコフ行列 $\mathbf{P}$ の定常分布 ${}^t\boldsymbol{\pi}$ を求めることができる [42]。べき乗法の原理は以下の通りである。

対角化可能な行列 $\mathbf{A}\in\mathbb{R}_{n\times n}$ が固有値

$$|\lambda_1|>|\lambda_2|\geq|\lambda_3|\geq\ldots\geq|\lambda_n|\geq 0$$

をもつとする（この場合、$\lambda_1$ は必ず実数である）。また、各固有値に対応する左固有ベクトルが ${}^t\mathbf{x}_1,{}^t\mathbf{x}_2,{}^t\mathbf{x}_3,\ldots,{}^t\mathbf{x}_n$ であるとする。

$$\begin{cases}{}^t\mathbf{x}_1\mathbf{A}=\lambda_1\,{}^t\mathbf{x}_1\\ {}^t\mathbf{x}_2\mathbf{A}=\lambda_2\,{}^t\mathbf{x}_2\\ {}^t\mathbf{x}_3\mathbf{A}=\lambda_3\,{}^t\mathbf{x}_3\\ \quad\vdots\\ {}^t\mathbf{x}_n\mathbf{A}=\lambda_n\,{}^t\mathbf{x}_n\end{cases}$$

$\mathbf{A}$ は対角化可能なので、初期ベクトル ${}^t\mathbf{v}$ を $\mathbf{A}$ の固有ベクトルの線形結合で表す。

$${}^t\mathbf{v}=v_1\,{}^t\mathbf{x}_1+v_2\,{}^t\mathbf{x}_2+\ldots+v_n\,{}^t\mathbf{x}_n\quad(v_1,v_2,\ldots,v_n\in\mathbb{R})$$

${}^t\mathbf{v}$ に $\mathbf{A}$ を $k$ 回乗ずると、

$$\begin{aligned}{}^t\mathbf{v}\mathbf{A}^k &= v_1\,{}^t\mathbf{x}_1\mathbf{A}^k + v_2\,{}^t\mathbf{x}_2\mathbf{A}^k + \cdots + v_n\,{}^t\mathbf{x}_n\mathbf{A}^k \\ &= v_1\lambda_1^k\,{}^t\mathbf{x}_1 + v_2\lambda_2^k\,{}^t\mathbf{x}_2 + \cdots + v_n\lambda_n^k\,{}^t\mathbf{x}_n \\ &= \lambda_1^k\left(v_1\,{}^t\mathbf{x}_1 + v_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\,{}^t\mathbf{x}_2 + \cdots + v_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\,{}^t\mathbf{x}_n\right)\end{aligned}$$

となる。$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n| \geq 0$ より、$\left(\frac{\lambda_2}{\lambda_1}\right)^k, \ldots, \left(\frac{\lambda_n}{\lambda_1}\right)^k$ の値は $k$ が大きくなるにつれて 0 に近づく。そのため、

$$\frac{1}{v_1}\,{}^t\mathbf{v}\left(\frac{1}{\lambda_1}\mathbf{A}\right)^k \to {}^t\mathbf{x}_1 \quad (k \to \infty) \tag{2.16}$$

となる。よって、絶対値最大の固有値 $\lambda_1$ に対応する固有ベクトル ${}^t\mathbf{x}_1$ を求めることができる。

条件を満たす確率行列の場合、絶対値最大の固有値は 1 である。そのため、条件を満たすマルコフ行列 $\mathbf{P}$ の絶対値最大の固有値に対応する固有ベクトルを求める場合、式（2.16）は

$${}^t\mathbf{v}\mathbf{P}^k \to {}^t\mathbf{x}_1 \quad (k \to \infty) \tag{2.17}$$

となる。

## 2.4 リンク解析

Web のリンク解析とは、Web のハイパーリンク構造を分析する手法である [35]。Web のハイパーリンク構造とは、Web コンテンツ（Web ページ、画像ファイルなど）をノードとし、それらをリンクによって接続して構成されるグラフである。このような有向グラフを Web グラフとよぶ。1998 年以後のほとんどすべての主要な検索エンジンは、Web ページの評価にあたってリンク解析をおこなっている [35]。

リンク解析は学術論文の引用分析 (citation analysis) をベースにしたものである。引用分析は「被引用数の多い論文は被引用数の少ない論文よりも重要な論文である」と考えて論文の相対的な重要度を測る手法である。引用分析を Web ページに応用して「被リンクの多い Web ページは被リンクの少ない Web ページよりも重要な Web ページである」と考えて Web ページを評価することが、リンク解析の出発点であろう。

1998 年、検索エンジン Teoma で使われている HITS アルゴリズム [34]、および検索エンジン Google で使われている PageRank アルゴリズム [48] の登場でリンク解析の分

野は大きく発展した[*30]。PageRank については次節以降でその詳細を述べる。以下で、HITS についての概略を述べる。

本稿では、Web ページ A に Web ページ B へのリンクがある場合、Web ページ A は（Web ページ B に）アウトリンク[*32]をしていると書き、Web ページ B は（Web ページ A による）インリンク[*33]をもつと書く。

アウトリンクとインリンクを区別する必要のない場合には、単にリンクとよぶ。あるリンクについて、アウトリンクをしている Web ページをリンク元の Web ページ、インリンクをもつ Web ページをリンク先の Web ページとよぶ。ある Web ページのインリンクの総数を調べるためには、すべての Web ページのアウトリンクを調べることになる [48]。

### 2.4.1 HITS

Brin らが PageRank アルゴリズムを考案したのと同じ 1998 年、Kleinberg は HITS（Hyperlink-Induced Topic Search）アルゴリズム [34] を発表した。HITS は PageRank 同様、Web のハイパーリンク構造を利用して Web ページの評価を行うアルゴリズムである。以下で、HITS を紹介する。

多くの Web ページへアウトリンクをしている Web ページをハブとよび、多くのインリンクをもつ Web ページをオーソリティとよぶ。それぞれの Web ページはハブスコアとオーソリティスコアをいう 2 つのスコアをもつ。HITS のハブとオーソリティの概念について、Kleinberg は次のように述べている [34]。

> a good hub is a page that points to many good authorities;
> a good authority is a page that is pointed to by many good hubs.

良質なハブの例としては良質なリンク集が挙げられる。また、良質なオーソリティの例としては、多くのリンク集からリンクを集めている、あるトピックに関して内容が充実している Web ページが挙げられる。

HITS と PageRank の違いは大きく分けて 2 つある [53]。HITS ではハブとオーソリティの両方の Web ページを重要な Web ページであると判断するが、PageRank の場合、ハブであることは評価せず、オーソリティにあたる Web ページのみを重要な Web ページであると判断する。

---

[*30] HITS は PageRank と同時期に発表されたが、2001 年に Teoma[*31]に採用されるまでは商用検索エンジンでは利用されていない [14]。

[*32] outlink。forward link、あるいは単にリンクとよぶこともある。

[*33] inlink。backlink、backward link、あるいは被リンクとよぶこともある。

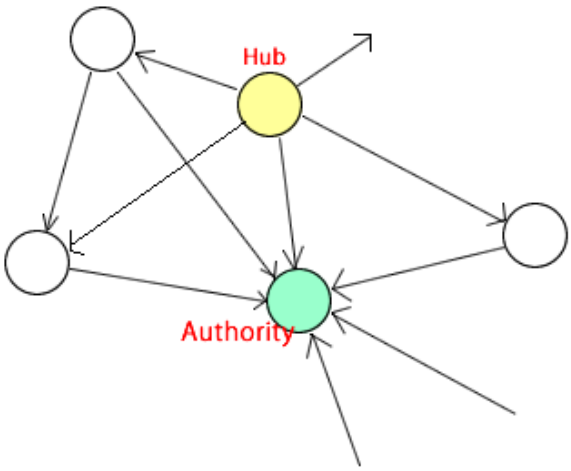


図 3　ハブとオーソリティ

また、HITS はクエリ従属なランキングアルゴリズムであり、PageRank はクエリ独立なランキングアルゴリズムである。クエリ独立なランキングアルゴリズムでは、Web ページの重要度のスコアの計算を予め行っておく。ユーザのクエリ入力に応じて重要度のスコアを参照し、Web ページに順位付けを行って検索結果を返す [35]。クエリ独立なランキングアルゴリズムの場合、インデックスした全ての Web ページの重要度のスコアを予め計算し、記憶しておく必要があるが、ユーザのクエリ入力を受けてから検索結果を返すまでの時間が短い[*34]。

一方、クエリ従属なランキングアルゴリズムでは、ユーザのクエリ入力を受けてから、クエリに応じた重要度のスコアを計算し、そのスコアを用いて検索結果を返す。重要度のスコアの計算には時間がかかる場合もあるため、クエリ応答時間が長くなる可能性がある。その一方で、ユーザの興味・関心に合わせて重要度のスコアを柔軟に変更することで、検索結果により多くの適切な Web ページを表示することができる可能性がある。

HITS についての詳細な紹介は本稿では省く。

[*34] Google は、「質の高い検索結果」の要素の 1 つに、クエリ応答時間の短さを挙げている [8]。

# 3 PageRank

本節では PageRank アルゴリズムおよび PageRank 値の考え方とその式としての表現、定義、計算方法を紹介する。

以下、単に PageRank といった場合、PageRank アルゴリズムという意味で使う。PageRank アルゴリズムとは、Web のハイパーリンク構造を利用して Web ページの相対的な重要度を算出し、Web ページを順位付けするアルゴリズムである。PageRank アルゴリズムで算出される Web ページの重要度を示す数値を PageRank 値とよぶ。

1998 年、Brin と Page は [48] および [8] で PageRank を発表した。Bianchini は、[48] と [8] とでは PageRank 値の定義のしかたがわずかに異なると述べている [6]。また Clausen は [8] の PageRank 値の定義に関する記述にはミスがあると指摘している [13]。PageRank に関連する研究では [48] と [8] の PageRank 値の考え方および定義を各自で解釈して、PageRank 値の再定義を行っているものがある [24][6][35]。

本節では PageRank 値の定義を紹介するにあたって、まず Page らによる PageRank 値の定義 [48] を紹介する。次に、Brin と Page による [8] の PageRank 値の定義を紹介し、[48] の PageRank 値の定義との相違点を紹介するとともに、Clausen が行った指摘を紹介する。また、PageRank に関連する研究者が行った PageRank 値の再定義の代表例として、Meyer らによる PageRank 値の定義を紹介する。次節以降では、Meyer らによる PageRank 値の定義をもとに話を進める。

## 3.1 PageRank 値の直感的な考え方

Page らは「PageRank 値の高い Web ページとは、インリンクの PageRank 値の合計が高い Web ページである。」と述べている [48]。PageRank 値は正の実数値であり、各 Web ページだけでなくリンクにも与えられる。各 Web ページがもつインリンクの PageRank 値の合計値が、各 Web ページの PageRank 値である。多くのインリンクをもつ Web ページ、および重要な Web ページからのインリンクをもつ Web ページを、重要な Web ページであると判断する。

インリンクの PageRank 値は、「リンク元の Web ページの PageRank 値」および「リンク元の Web ページのアウトリンクの総数」で計算される。例えば、Yahoo!のように高い PageRank 値の Web ページからの 1 本のインリンクは、内容の乏しい Web ページからの複数のインリンクの PageRank 値の合計よりも高い PageRank 値をもつ。また、同じ

PageRank 値の Web ページからのアウトリンクでも、その Web ページのアウトリンクの総数が多いほど 1 本あたりのアウトリンクがリンク先の Web ページに与える PageRank 値は低くなり、アウトリンクの総数が少ないほど 1 本あたりのアウトリンクがリンク先の Web ページに与える PageRank 値は高くなる。

ただし、自分自身へのアウトリンクは PageRank 値の考慮には入れない。また、Web ページ A から Web ページ B へ 2 つ以上のアウトリンクがある場合でも 1 つのアウトリンクとして考える。

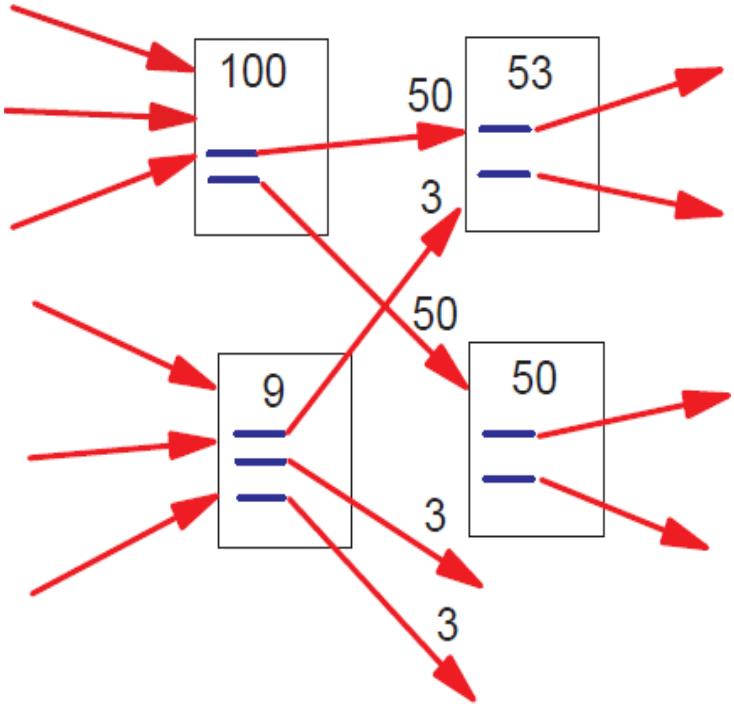


図 4 PageRank 値の考え方 （[48] より引用）

図 4 は Page らが PageRank 値の考え方を説明する際に用いた図である [48]。「リンク元の Web ページの PageRank 値」が 100 で、「リンク元の Web ページのアウトリンクの総数」が 2 本である場合、1 本あたりのアウトリンクがリンク先の Web ページに与える PageRank 値は 50 である。

## 3.2 ランダムサーファーモデル

Page らは PageRank 値の考え方をランダムサーファーモデルで説明している [48]。

ランダムサーファーとは、Web グラフを遷移していくユーザを抽象化したものである。ランダムサーファーの挙動（遷移のしかた）は、現在のノードがダングリングノード (dangling node) であるか非ダングリングノードであるかによって異なる。ダングリングノードとはアウトリンクを 1 つももたない Web ページのこと[*35]であり、非ダングリングノードとはダングリングノードでない Web ページのことである。

ランダムサーファーの挙動は以下の通りである[*36]。

- 現在のノードがダングリングノードであれば、挙動 1 を選択する。

  [挙動 1] 遷移を終了するか、ランダムに選んだノードに遷移する。

- 現在のノードが非ダングリングノードであれば、挙動 2 または挙動 3 を選択する。

  [挙動 2] 現在のノード内のアウトリンクを辿って遷移する。

  [挙動 3] 現在のノード内のアウトリンクと関係なく、ランダムに選んだノードに遷移する。

ランダムサーファーには、元の Web ページへ戻る[*37]（現在の Web ページにアウトリンクがないにも関わらず、1 つ前にいた Web ページへ遷移する）挙動は想定されていない。また、自分自身へのアウトリンクは PageRank 値の考慮に入れないため、現在の Web ページに滞在する（現在の Web ページから現在の Web ページへ遷移する）挙動も想定されていない（ただし、挙動 3 でたまたま現在のページに遷移することはある。）。

Page らは挙動 3 の身近な例として、サーファーが Web ページのリンクを辿ることに飽きてブックマーク（お気に入り）に登録してある Web ページへ遷移したり、ブラウザの URL 欄に目的の Web ページの URL を直接入力して遷移する場合が挙げられると述べている [48]。

---

[*35] ダングリングノードに該当する Web コンテンツとしては、アウトリンクを 1 つももたない Web ページ、画像ファイル、PDF ファイルなどがある。また Page らは、URL サーバによるクローラの巡回命令は出されているが、その後のクロールおよびインデックスが行われていない Web コンテンツについても、ダングリングノードとして処理している [48]。

[*36] Brin らは [48] でダングリングノードに遷移した場合のランダムサーファーの挙動については述べておらず、ランダムサーファーの挙動としては挙動 2 と挙動 3 しか挙げていない。しかし、本稿ではランダムサーファーの挙動を詳しく述べるために 3 つの挙動に分ける。

[*37] 現実のユーザはインターネットブラウザの「戻る」ボタンを使うことがあるが、1998 年に発表された PageRank の場合、「戻る」ボタンを考慮に入れていない。「戻る」ボタンを考慮に入れた PageRank については 4.1.4 節で述べる。

ランダムサーファーはランダムに選んだノードから遷移を開始し、上記の挙動を繰り返す。ランダムサーファーが繰り返し遷移する Web ページはインリンクの PageRank 値の合計が高い Web ページであり、相対的に重要な Web ページであると考えられる。それに対して、ランダムサーファーがほとんど遷移しない Web ページはインリンクの PageRank 値の合計が低い Web ページであり、相対的に重要でない Web ページであると考えられる。ランダムサーファーが上記の 3 つの挙動を無限回繰り返す中で、各 Web ページへどれだけ遷移したかの頻度を示すランダムサーファーの存在確率が PageRank 値である。

ランダムサーファーの遷移は現在の Web ページにのみ依存するため、ランダムサーファーの遷移確率はマルコフ連鎖の遷移確率行列で表すことができる。ランダムサーファーの遷移確率行列は primitive なマルコフ行列であるから定常分布が存在し、べき乗法によりその定常分布を求めることができる（3.7 節参照)。

## 3.3 Simplified PageRank

Page らは PageRank 値の考え方を式で表している [48]。それにより定義される PageRank 値を本稿では Simplified PageRank 値とよぶ。Simplified PageRank 値の式は以下のように表される。

$W$ を対象とする Web ページ全体の集合とする。$u \in W$ のとき、$F_u$ で Web ページ $u$ のリンク先の Web ページの集合を表し、$B_u$ で Web ページ $u$ のリンク元の Web ページの集合を表す。また $n_u$ で Web ページ $u$ のアウトリンクの総数を表す（$n_u = |F_u|$)。

このとき以下の方程式を満たす解 $\{r_u\}$ を考える。$r_u$ をページ $u$ の PageRank 値という。

$$r_u = c \sum_{v \in B_u} \frac{r_v}{n_v} \quad \text{for all } u \tag{3.1}$$

ただし $c$ はすべての Web ページの PageRank 値の合計を一定の値 $r_{\text{all}}$ $(> 0)$ にするための係数である[*38]。$r_{\text{all}}$ の値は PageRank 値の値を変化させるが、PageRank 値の相対的な大きさによって Web ページの順位付けを行う用途には影響しない。以下では、$W = \{1, 2, \ldots, n\}$ として Web ページを数字で表す。

Page らは、同論文で式 (3.1) をベクトルと行列とを使って表現している。以下のような

[*38] $c$ は decay factor とよばれている [48]。

正方行列 $\mathbf{H}$ で Web のハイパーリンクの構造を表し、それをハイパーリンク行列とよぶ。

$$h_{uv} = \begin{cases} \frac{1}{n_u} & (u \text{ から } v \text{ へのアウトリンクが存在する場合}) \\ 0 & (u \text{ から } v \text{ へのアウトリンクが存在しない場合}) \end{cases}$$

ハイパーリンク行列は、ダングリングノードの行ならば行の成分はすべて 0 となり、非ダングリングノードの行ならば行の成分の合計は 1 となる。

また、$\mathbf{r}$ を各 Web ページの PageRank 値を成分とする列ベクトルとし、PageRank ベクトルとよぶ。この設定で式（3.1）を書き直すと、

$$^t\mathbf{r} = c\, ^t\mathbf{r}\mathbf{H} \tag{3.2}$$

となる[*39]。式 (3.1) または式 (3.2) で表される PageRank 値を Simplified PageRank 値とよぶ。

以下では、Web グラフ上に非ダングリングノードが存在するとする。すると $\mathbf{H} \neq \mathbf{0}$ である。行列 $\mathbf{H}$ は非負行列であるから、Perron-Frobenius の定理の非負行列への拡張 ([42] §8.3) により $\rho(\mathbf{H}) \in \sigma(\mathbf{H})$ であり、非負の実ベクトルが $\rho(\mathbf{H})$ に対応する固有ベクトルとなる。また、$\rho(\mathbf{H}) = 0 \Leftrightarrow \mathbf{H} = \mathbf{0}$ なので $c = \frac{1}{\rho(\mathbf{H})}$ とおくと、$^t\mathbf{r} = c\, ^t\mathbf{r}\mathbf{H}$ を満たす非自明な非負の実ベクトル $^t\mathbf{r}$ が存在する。したがってこの場合は式（3.2）に非自明な解が必ず存在するが、以下の 3 つの問題がある。

(a) 一部の Web ページの PageRank 値が 0 になってしまう場合がある。これは、それらの Web ページ間で相対的な順位をつけることができないことを意味する。
(b) ランダムサーファーモデルで考えた場合に、ランダムサーファーの存在確率が収束しない場合がありえる。
(c) 収束したとしても定常分布が一意的に定まるとは限らない。

(a) の問題は、Web グラフ上にランクシンク（rank sink[*40]) が存在すると起きる。(b) の問題は、次ページで定義するループが存在すると起きる。(c) の問題は、Web グラフが強連結でないときに起きる場合がある。

ランクシンクとは、以下で定義する Web ページの集合である。$W$ を対象とする Web ページ全体の集合とする。$A \subset W$ が次の性質を満たしているときに $A$ は（狭義の）ランクシンクであるという。

---

[*39] Page らの論文では式 (3.2) は本稿の記法を用いると $\mathbf{r} = c\, ^t\mathbf{H}\mathbf{r}$ と書かれているが、本稿では後に再定義する PageRank 値と同じ書き方に統一するために、$^t\mathbf{r} = c^t\mathbf{r}\mathbf{H}$ と書く。以後の引用箇所においても同様である。

[*40] PageRank 値の溜まり場、というニュアンスであると考えられる。sink には台所の流しなどの意味がある。

(i) $A \neq W$

(ii) $W - A$ に属するある Web ページから $A$ に属する Web ページへのアウトリンクはあるが、$A$ のどの Web ページからも $W - A$ のどの Web ページへのアウトリンクがない。

(iii) $A$ は強連結である。

(ii) から、$A$ は $\phi$ ではない。

ランクシンクには、広義のランクシンクと狭義のランクシンクがある。広義のランクシンクは、上記の性質 (i)(ii) の 2 つの性質を満たす $W$ の部分集合を指す[*41]。狭義のランクシンクは、上記の性質 (i)(ii)(iii) をすべて満たす $W$ の部分集合を指す[*42]。広義のランクシンクだが狭義のランクシンクではない Web ページの集合としては、1 つのダングリングノードからなる集合がある。本稿では以後、単にランクシンクといった場合には狭義のランクシンクを意味する。

Web グラフ中に（狭義の）ランクシンクが 1 つでも存在する場合、ランクシンクに属する Web ページは、他の Web ページからのインリンクをうけて PageRank 値を溜め込んだまま、ランクシンクに属さない Web ページへ PageRank 値を配分しない。その結果、ランクシンクに属する Web ページの PageRank 値だけが高まり、ランクシンクに属さず、リンクを辿ってランクシンクへ到達できる Web ページの PageRank 値は 0 になってしまう。これを PageRank 値の溜めこみ（Hoarding）の問題という。

(b) の問題は、2 つかそれ以上の Web ページが環状にアウトリンクをもち、他の Web ページへのアウトリンクをもたない場合に起きる。このような Web ページの集合はループ (loop)[*43]に分類される。

正確にはループは以下のように定義される Web ページあるいはグラフのノードの集合である。以下、Web ページとして定義する。$W$ を対象とする Web ページ全体の集合とする。$W$ もしくはその部分集合が、$r$ 個の部分集合 $D_1,\ D_2,\ \cdots,\ D_r$ に分割されているとする。$D_1$ の中の Web ページからのアウトリンクが、

$$D_1 \to D_2 \to \cdots \to D_r \to D_1 \to \cdots \tag{3.3}$$

の順番に各部分集合の中の Web ページを指し、他のアウトリンクが存在しないような $W$ とする。このとき $D_1, \ldots, D_r$ をループとよぶ。

[*41] [35] ではこの意味でランクシンクという言葉を使っている。

[*42] [48] ではこの意味でランクシンクという言葉を使っている。

[*43] 循環 (cycle) ともよぶ [35]。

ループが存在する場合、初期分布によっては (b) の問題が生じ、ランダムサーファーの存在確率は収束せず、振動する。この場合には 2.3.9 節のべき乗法で PageRank 値を常に算出できるとは限らない。周期的なマルコフ連鎖のマルコフ行列 **A** の有向グラフにはループが存在する。また、ループが存在する既約なマルコフ行列 **A** は周期的である。

また (c) の問題は、例えば Web グラフが 2 つの強連結部分グラフから成り、2 つの部分グラフ間にはリンクが全くない場合に生じる。この場合には 2 つの部分グラフ間に属する Web ページ間で相対的な評価を行うことができない。

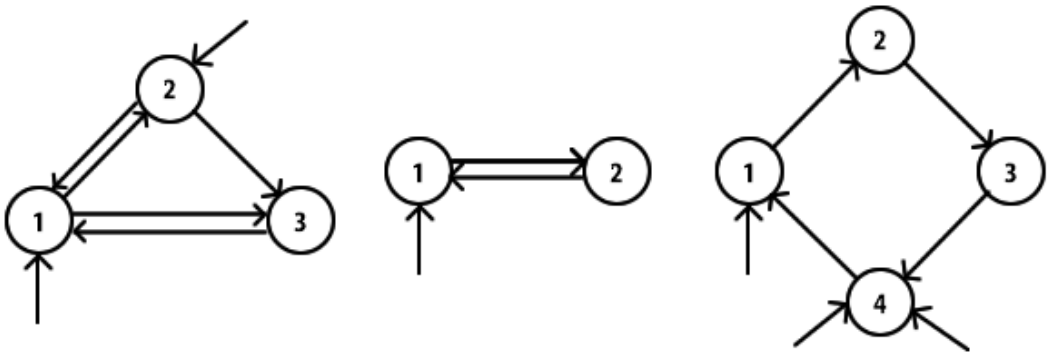


図 5 (左）ランクシンクの例 （中・右）ループの例

図 4.1.2 は 6 つのノードからなる Web グラフを示したものである。図 4.1.2 のハイパー

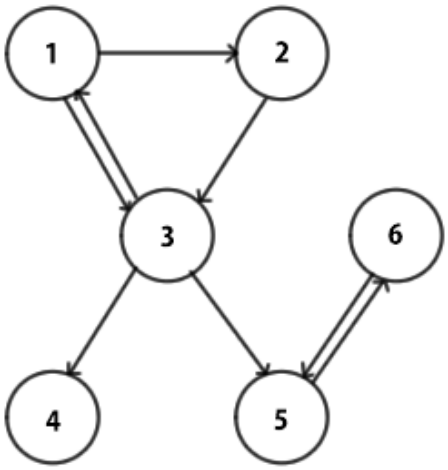


図 6 Web グラフの例

リンク構造を行列 **H** で表現すると、次のようになる。

$$\mathbf{H} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{3.4}$$

図 4.1.2 では、Web ページ 5 と Web ページ 6 からなる集合はループである。Web ページ 1，2，3 からなる集合はループではない。また Web ページ 4 はダングリングノードである。そのため行列 **H** のノード 4 の行の成分はすべて 0 となる。このように行列 **H** は、ダングリングノードの行の成分の合計が 0 となり、非ダングリングノードの行の成分の合計が 1 となる。

現実の Web にはランクシンク、ダングリングノード、ループが存在する。Page らはこれらによって生じる、PageRank 値の溜めこみ等の 3 つの問題を克服すべく、Simplified PageRank 値に変更を加えた。

## 3.4 Normal PageRank

Page らは PageRank 値の溜めこみ等の 3 つの問題を克服するために、ランクソース（rank source）を導入している [48]。

ランクソースとは、Web グラフ上の全ての Web ページに与える一定の PageRank 値のことである。ある Web ページの PageRank 値はインリンクの PageRank 値の合計を示すものであるから、ランクソースを導入することで全ての Web ページに一定の PageRank 値を与えた場合、1 つもインリンクをもたない Web ページでも、ランクソースにより与えられた PageRank 値に相当するインリンクをもつと解釈できる。

ランクソースを導入した PageRank 値を Simplified PageRank 値に対して本稿では Normal PageRank 値とよぶ。Normal PageRank 値を以下のように定義する。

**e** で各 Web ページに配分するランクソースの値を成分とする列ベクトル（ランクソースベクトル）を表す。Page らは $||\mathbf{e}||_1 = 0.15$ を推奨している [48]。$e_u$ で Web ページ $u$ に配分されるランクソースを表し、$e_u = \alpha$ とする $(0 < \alpha \leq 1)$。$\mathbf{r}'$ で Normal PageRank 値を成分とする Normal PageRank ベクトルを表す。Web ページ $u$ の PageRank 値 $r'_u$

を、

$$r'_u = c \sum_{v \in B_u} \frac{r'_v}{n_v} + ce_u \tag{3.5}$$

と定義する。ただし $c$ は式（3.5）の解 $\mathbf{r}'$ が $||\mathbf{r}'||_1 = 1$ を満たすための係数 $(c > 0)$ である。

式 (3.5) の行列表現は ${}^t\mathbf{r}' = c({}^t\mathbf{r}'\mathbf{H} + {}^t\mathbf{e})$ である。この式は、$||\mathbf{r}'||_1 = 1$ より、

$${}^t\mathbf{r}' = c\,{}^t\mathbf{r}'(\mathbf{H} + \mathbf{1}\,{}^t\mathbf{e}) \tag{3.6}$$

と変形できる（$\mathbf{1}$ はすべての成分が 1 の列ベクトル）。式 (3.6) は ${}^t\mathbf{r}'$ が行列 $(\mathbf{H} + \mathbf{1}\,{}^t\mathbf{e})$ の固有値 $\frac{1}{c}$ に対応する左固有ベクトルであることを意味している。

$\mathbf{H}' = \mathbf{H} + \mathbf{1}\,{}^t\mathbf{e}$ とおくと、$\mathbf{H}' > 0$ であり、これは既約な行列である。2.3.6 節の Perron-Frobenius の定理 ([42] §8.3) により、$\rho(\mathbf{H}') \in \sigma(\mathbf{H}') \wedge 0 < \rho(\mathbf{H}')$ であり、一意的な左 Perron ベクトル ${}^t\mathbf{r}'$ $({}^t\mathbf{r}' > 0$ かつ $||{}^t\mathbf{r}'||_1 = 1)$ が存在し、$\rho(\mathbf{H}')\,{}^t\mathbf{r}' = {}^t\mathbf{r}'\mathbf{H}'$ となる。したがって、$c = \frac{1}{\rho(\mathbf{H}')}$ とすれば、式 (3.6) の解は一意的に存在する。また (a)、(b)、(c) の問題が解消されているのもわかる。

## 3.5 PageRank 値の定義の訂正

本小節では Brin と Page によるもう 1 つの論文 [8] の PageRank 値の定義を紹介する。また、Clausen による [8] の PageRank 値の定義の誤りの指摘 [13] を紹介する。

[8] の PageRank 値の定義は以下の通りである。$d$ を damping factor とよび、$0 \leq d < 1$ とする。Web ページ $u$ の PageRank 値 $r'_u$ を以下で定義する。

$$r'_u = d \sum_{v \in B_u} \frac{r'_v}{n_v} + (1 - d) \tag{3.7}$$

$d$ は現在のノード内のリンクを辿って遷移する割合と、ランダムに選んだノードに遷移する割合を決定するパラメータである。

Page らは $d$ の値について、$d = 0.85$ を推奨している。また式（3.7）について、

> Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.

と説明している [8]。

Clausen は、式（3.7）とその説明（引用箇所）には誤りがあると指摘している [13]。式（3.7）の右辺第 2 項が $(1 - d)$ であると、全ての Web ページの PageRank 値の合計値は

1 にはならない。全ての Web ページの PageRank 値の合計を 1 にするためには、右辺第 2 項を $\frac{(1-d)}{n}$（$n$ は Web ページの総数）にする必要がある。

Brin と Page による [48] と [8] とでの PageRank 値の定義の相違、および [8] での定義の誤りについて、Clausen のように陽に指摘している論文は少ない。PageRank に関連する研究を行っている研究者達は、[8] での PageRank 値の定義は誤り (mistake) であるという書き方は控え、[48] および [8] での PageRank 値の定義の意図を解釈して PageRank を再定義したものを、Brin と Page らの定義した PageRank 値として記述している [24][6]。PageRank 値を再定義した例として、(3.7) 節で Meyer らの定義を紹介する。

## 3.6 べき乗法による PageRank 値の計算

Page らは Normal PageRank 値の式 (3.6) を用いて Normal PageRank ベクトルをべき乗法で計算している [48]。初期ベクトル $\mathbf{s}$ は任意のベクトルである。

$$
\begin{aligned}
{}^t\mathbf{r}'_0 &\leftarrow {}^t\mathbf{s} \\
\text{loop} &: \\
{}^t\mathbf{r}'_{i+1} &\leftarrow {}^t\mathbf{r}'_i\mathbf{H} \\
d &\leftarrow ||{}^t\mathbf{r}'_i||_1 - ||{}^t\mathbf{r}'_{i+1}||_1 \\
{}^t\mathbf{r}'_{i+1} &\leftarrow {}^t\mathbf{r}'_{i+1} + d\,{}^t\mathbf{e} \\
\delta &\leftarrow ||{}^t\mathbf{r}'_{i+1} - {}^t\mathbf{r}'_i||_1 \\
\text{while } &\ \delta > \epsilon
\end{aligned}
$$

$d$ は $||{}^t\mathbf{r}'||_1$ の値を維持するために用いる変数である。例えば行列 $\mathbf{H}$ が正行列で各行の合計値が 1 以下であり、$\mathbf{e}$ が正の確率ベクトルであれば ${}^t\mathbf{r}'$ は収束する。

Page らは同論文で、322,000,000 個の Web ページのリンクのデータベースに対して、べき乗法により PageRank 値の算出を行った場合は、52 回の反復で収束したと述べている。また、半分の 161,000,000 個の Web ページの場合は、45 回の反復で収束したと述べている。

Page らは [48] で、ダングリングリンク（dangling link）を除外して PageRank 値を算出すると述べている。ダングリングリンクとは、ダングリングノードへのアウトリンクのことである[*44] 。しかし、ダングリングリンクをどのように除外するかについての詳細な

[*44] [48] ではダングリングリンクは次のように定義されている。
"Dangling links are simply links that point to any page with no outgoing links."

記述は必ずしも行われていない。Brin と Page はダングリングリンクの除外について論文 [48] では次のように述べている。

> Because dangling links do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated. After all the PageRanks are calculated, they can be added back in, without affecting things significantly. Notice the normalization of the other links on the same page as a link which was removed will change slightly, but this should not have a large effect.

また Brin と Page は別の論文 [9] では次のように述べている。

> As a solution, we often remove nodes with no outgoing edges during the computation of PageRank, then add them back in after the weights have stabilized.

ダングリングリンクを除外する方法には少なくとも 2 通りあると考えられる。例えば Web ページ A がダングリングリンクしかアウトリンクをもたない場合、Web ページ A のもつダングリングリンクを除外すると、Web ページ A は新たにダングリングノードになる。すると、Web ページ A へのアウトリンクしかもたない Web ページ B は、ダングリングリンクしかもたない Web ページになる。このとき、

(a)Web ページ B のダングリングリンクは除外しない。
(b)Web ページ B のダングリングリンクも再帰的に除外する。

の 2 通りの方法が考えられる。(a) の場合、Web ページ B は非ダングリングノードのままである。しかし (b) の場合、Web ページ B はダングリングノードになる。(b) の方法では全ての Web ページが非ダングリングノードになるまで、ダングリングリンクの除外が繰り返し行われる。理論上は、全ての Web ページのアウトリンクが除外されてしまう可能性もある。(a) を採用するか (b) を採用するかによって、PageRank 値の計算の際に考慮に入れるリンクは変化する。

(a) の場合、ダングリングリンクの定義は次のように書き換えられる。

**定義 A** 「ダングリングリンクとは、ダングリングノードへのアウトリンクのことである。ただしダングリングリンクを除外した結果として現れたダングリングリンクは、ダングリングリンクに含めない。」

(b) の場合、ダングリングリンクの定義は次のように書き換えられる。

**定義 B** 「ダングリングリンクとは、何回かリンクを辿ると必ずダングリングノードに至るアウトリンクのことである。」

おそらく、Page らは定義 A の意味でダングリングリンクという言葉を使っていると考えられるが、読者によっては定義 B の意味でダングリングリンクという言葉を解釈する可能性も無いとは言えない。

ダングリングリンクを除外して PageRank 値の計算を行うことには問題があるという論文も発表されている。McCurley らは、ダングリングノードのほうが非ダングリングノードよりも順位が高い場合があることがあることを示した [16]。ダングリングノードを除外した場合としない場合とで、Web ページの順位付けが変わってしまう場合もありうる。複数の研究グループが、ダングリングノードを除かずに PageRank 値の計算を行う手法の提案を行っている [16][39][37]。

## 3.7 Meyer らによる PageRank 値の定義

Meyer らは、[35] で Page らとは別の書き方で PageRank 値の定義を行っている。本小節では Meyer らによる Simplified PageRank 値の修正を紹介する。

Meyer らは、Simplified PageRank 値のハイパーリンク行列 $\mathbf{H}$ に対して 2 つの修正を行っている。2 つの修正を stochasticity adjustment と primitivity adjustment とよぶ。

stochasticity adjustment では、$n \times n$ 行列 $\mathbf{H}$ のダングリングノードの行のすべての成分を $\frac{1}{n}$ で置き換えることで、ダングリングノードが存在する場合でも行列 $\mathbf{H}$ を確率行列となるように修正する。stochasticity adjustment を行ったハイパーリンク行列を $\mathbf{S}$ と表し、以下のように定義する。

$$\mathbf{S} = \mathbf{H} + \mathbf{a}\left(\frac{1}{n}{}^t\mathbf{1}\right) \tag{3.8}$$

ただし、$\mathbf{a}$ は列ベクトルで、その成分は以下の通りである（$\mathbf{a}$ をダングリングノードベクトルとよぶ）。

$$a_i = \begin{cases} 1 & (Web\,\text{ページ}\,i\,\text{がダングリングノード}) \\ 0 & (Web\,\text{ページ}\,i\,\text{が非ダングリングノード}) \end{cases}$$

行列 $\mathbf{S}$ は確率行列となるので、$\rho(\mathbf{S}) = 1$ である。

図 4.1.2 のハイパーリンク構造を行列 **S** で表現すると、次のようになる。

$$\mathbf{S} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

primitivity adjustment では、べき乗法が収束することを保証するために行列 **S** を primitive な行列に修正している。行列 **S** に対して primitive adjustment を行った行列を $\mathbf{G}$[*45]とよぶ。

$$\mathbf{G} = d\mathbf{S} + (1-d)\frac{1}{n}\,\mathbf{1}\,{}^t\mathbf{1} \tag{3.9}$$

$$(\,= d\mathbf{H} + (d\mathbf{a} + (1-d)\mathbf{1})\frac{1}{n}\,{}^t\mathbf{1}) \tag{3.10}$$

ただし $d$ は damping factor である (式 (3.7))。

Meyer らの PageRank 値の定義では、ランダムサーファーの挙動は以下の通りである。

- 現在のノードがダングリングノードであれば、挙動 1 を選択する。

  [挙動 1] 全ての Web ページからランダムに選んだノードに遷移する。

- 現在のノードが非ダングリングノードであれば、挙動 2 あるいは挙動 3 を選択する。

  [挙動 2] 現在のノード内のアウトリンクを辿って遷移する。

  [挙動 3] 現在のノード内のアウトリンクと関係なく、ランダムに選んだノードに遷移する。

現在のノードがダングリングノードであれば、$d$ の値に関わらず全ての Web ページに対して $\frac{1}{n}$ の確率で遷移する。この場合、自分自身に再び遷移することもありうる。また、現在のノードが非ダングリングノードであれば、$d = 0.6$ の場合、60% の割合で挙動 2 を選択し、40% の割合で挙動 3 を選択する。式 (3.10) から、Page らのランクソースを Meyer らが $(d\mathbf{a} + (1-d)\mathbf{1})\frac{1}{n}\,{}^t\mathbf{1})$ と解釈していることがわかる。

[*45] Meyer らはこの行列を Google 行列とよんでいる。

図 4.1.2 のハイパーリンク構造を行列 $\mathbf{G}$ で表現すると、次のようになる。ただし、$d = \frac{9}{10}$ とした場合である。

$$\begin{aligned}\mathbf{G} &= \frac{9}{10}\mathbf{S} + \frac{1}{10} \times \frac{1}{6}\ \mathbf{1}\ {}^t\mathbf{1} \\ &= \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1/60 & 28/60 & 28/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 55/60 & 1/60 & 1/60 & 1/60 \\ 19/60 & 1/60 & 1/60 & 19/60 & 19/60 & 1/60 \\ 10/60 & 10/60 & 10/60 & 10/60 & 10/60 & 10/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 55/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 55/60 & 1/60 \end{pmatrix} \end{array}\end{aligned} \tag{3.11}$$

$\mathbf{G}$ は正かつ既約な確率行列で、$\mathbf{G}^1 > 0$ であるから $\mathbf{G}$ は primitive な行列である。そのため、$\mathbf{G}$ のべき乗は収束することが保証され、べき乗法によりマルコフ行列 $\mathbf{G}$ の最大固有値 1 に対応する固有ベクトル ${}^t\mathbf{r}'$（PageRank ベクトル）を求めることができる。解は以下を満たす一意的なベクトルである。

$${}^t\mathbf{r}' = {}^t\mathbf{r}'\mathbf{G} \quad \text{かつ} \quad {}^t\mathbf{r}'\mathbf{1} = 1 \tag{3.12}$$

実際の計算にあたっては、以下のように密な行列 $\mathbf{G}$ ではなく疎な行列 $\mathbf{H}$ を使って反復計算を行うことができる（${}^t\mathbf{r}'^{(k)}$ は $k$ 回目の反復時の PageRank ベクトルを表す）。

$$\begin{aligned}{}^t\mathbf{r}'^{(k+1)} &= {}^t\mathbf{r}'^{(k)}\mathbf{G} \\ &= d\ {}^t\mathbf{r}'^{(k)}\mathbf{S} + (1-d)\ {}^t\mathbf{r}'^{(k)}\frac{1}{n}\ \mathbf{1}\ {}^t\mathbf{1} \\ &= d\ {}^t\mathbf{r}'^{(k)}\mathbf{H} + (d\mathbf{a}\ {}^t\mathbf{r}'^{(k)} + (1-d)\ {}^t\mathbf{r}'^{(k)}\mathbf{1})\frac{1}{n}\ {}^t\mathbf{1} \\ &= d\ {}^t\mathbf{r}'^{(k)}\mathbf{H} + (d\mathbf{a}\ {}^t\mathbf{r}'^{(k)} + 1 - d)\frac{1}{n}\ {}^t\mathbf{1}\end{aligned} \tag{3.13}$$

Meyer らは、PageRank 値の計算にべき乗法が使われている理由として、次の 4 つを挙げている [35]。

- 実装がシンプルである。
- 計算にあたっては、密な行列 $\mathbf{G}$ ではなく疎な行列 $\mathbf{H}$ で計算を行うことができる。
- 記憶すべき要素は以下の 3 つだけである。
  - ハイパーリンク行列 $\mathbf{H}$
  - ダングリングノードベクトル $\mathbf{a}$
  - $k$ 回目の反復の際の PageRank ベクトル ${}^t\mathbf{r}'$

- 収束までにかかる反復数が少ない。

以下は、式 (3.11) の $\mathbf{G}$ の定常分布をべき乗法で求めた PageRank ベクトル ${}^t\mathbf{r}$ である（収束判定条件 $\epsilon = 10^{-8}$ で計算を行った）。

$$ {}^t\mathbf{r} = (0.05160 \quad 0.04765 \quad 0.09057 \quad 0.05160 \quad 0.38644 \quad 0.37214) $$

6 つの Web ページの PageRank は高い順に、5、6、3、1、4、2 である（1 と 4 の順位は同じ）。

Meyer らは、Page らの論文 [48] でマルコフ連鎖という言葉が一度も使われていないことについて、1998 年の時点では Page らがマルコフ連鎖の概念を知らなかったことが理由ではないかと述べている。Simplified PageRank 値の修正にあたって、Page らはランダムサーファーモデルを使い、Meyer らはマルコフ連鎖の概念を使っているため、両者に違いが出た可能性がある。

# 4 PageRank の改良

本節では PageRank の改良研究を紹介する。PageRank に関する研究は全世界で盛んに行われているが、本稿で紹介する研究はその一部である。本稿では PageRank の改良研究を 4 つのテーマ：

- 4.1 節：PageRank 値の改良による検索結果の質の向上
- 4.3 節：PageRank 値の計算にかかる記憶領域の節約
- 4.4 節：PageRank 値の計算の高速化
- 4.5 節：PageRank 値の更新の高速化

に分類し、それぞれのテーマについての代表的な改良研究を紹介する。また、これらの研究の理解に必要となる連立一次方程式（線形方程式系）による PageRank 値の計算を 4.2 節で紹介する。

以下、Brin と Page が 1998 年に発表した PageRank を通常の PageRank とよぶ。また、PageRank 値の定義およびランダムサーファーの挙動には Meyer らの記述 (3.7 節) を用いる。その際、PageRank ベクトルを $\mathbf{r}'$ ではなく $\mathbf{r}$ と表す。

## 4.1 PageRank 値の改良による検索結果の質の向上

PageRank の改良テーマの 1 つに、PageRank 値の改良による検索結果の質の向上がある。検索結果の質を向上させる手法として、4.1.1 節でパーソナライズド（personalization）ランクソース、4.1.2 節で知的サーファーモデル、4.1.3 節で damping factor によるリンクスパムの特定、4.1.4 節でバックボタンモデルを紹介する。

### 4.1.1 パーソナライズドランクソース

Page らは、ランクソースを導入してランクシンクによる PageRank 値の溜めこみの問題を解消した [48]。Page らは同論文で、個々のユーザの興味・関心に合わせてランクソースに変化を加えることで、個々のユーザの興味・関心に合わせた PageRank 値（パーソナライズド PageRank 値）を算出して検索結果の質を向上させることを今後の課題として掲げている[*46]。以下、パーソナライズド PageRank 値を算出する PageRank アルゴリズムをパーソナライズド PageRank とよぶ。

通常の PageRank では全ての Web ページに同一のランクソースを与えるのに対し、パーソナライズド PageRank では、ユーザの興味・関心に応じてランクソースベクトル $\mathbf{e} = \frac{1}{n}{}^t\mathbf{1}$ の成分を変更する [35]。$\mathbf{e}$ の成分を変更したベクトルをパーソナライズドランクソースベクトルとよび $\mathbf{v}$ と表す。$\mathbf{v}$ は $||\mathbf{v}||_1 = 1$ の確率ベクトルである。ランクソースベクトル $\mathbf{e}$ をパーソナライズドランクソースベクトル $\mathbf{v}$ に置き換えた場合、式 (3.9) の行列 $\mathbf{G}$ は以下のようになる。

$$\mathbf{G} = d\mathbf{S} + (1-d)\ \mathbf{1}\ {}^t\mathbf{v} \tag{4.1}$$

パーソナライズドランクソースベクトルを使った PageRank に関連する代表的な研究としては、Haveliwala による Topic-Sensitive PageRank がある [24][25]。以下ではその概要を紹介する。

通常の PageRank では、単一の PageRank ベクトル $\mathbf{r}$ を用いて Web ページの順位付

[*46] 2005 年 11 月 11 日、Google は「パーソナライズド検索」サービスの $\beta$ 版の提供を開始した。ユーザの興味・関心に合わせた検索結果を提供するためにユーザの「ウェブ履歴」を記録することで、
- （過去に）閲覧したウェブページや Google で実行した検索を表示、検索する。
- 最もよく利用したサイトや上位の検索キーワードなどのウェブ利用状況に関連する統計情報を確認する。
- 過去に実行した検索や閲覧したサイトに基づき、検索結果をパーソナライズする。

ことができると述べている (箇条書きの内容は [20] より引用)。

けを行う。それに対して Topic-Sensitive PageRank では、ODP[*47]のカテゴリの第 1 階層の 16 種類のトピック[*48]それぞれに関して PageRank ベクトル $\mathbf{r}_i$ $(i = 1, 2, \ldots, 16)$ を算出し、ユーザのクエリ入力に応じてそれらを組み合わせて 1 つの PageRank ベクトル $\mathbf{r}$ を生成している。

$$\mathbf{r} = \beta_1 \mathbf{r}_1 + \beta_1 \mathbf{r}_1 + \cdots + \beta_{16} \mathbf{r}_{16} \tag{4.2}$$

ただし、$\sum_i \beta_i = 1$ とする。$\beta_i$ の値は、入力クエリが 16 種類のどのトピックに関連する単語に該当するかを判定して決定される [25]。

Haveliwala は以下のようなアンケートを実施し、Topic-Sensitive PageRank による検索結果の質の向上を検証している。5 人のボランティアが前もって決定しておいた 10 個のクエリで検索を行い、通常の PageRank による検索結果と Topic-Sensitive PageRank による検索結果とではどちらが適切な検索結果を表示しているかを回答している。データセットには、スタンフォード大学の WebBase[*49]の約 120,000,000 の Web ページを利用している。アンケートの結果、10 個のクエリのうち 8 個のクエリについて、Topic-Sensitive PageRank のほうが適切な検索結果を表示していると答えたボランティアが過半数を占めたと述べている。Haveliwala は、今後の課題として ODP のカテゴリの第 2 階層、第 3 階層などのより細かいトピックの分類を用いて、検索結果の質を向上させることを課題として挙げている。

通常の PageRank がクエリ独立なランキングアルゴリズムであるのに対し、Haveliwala の Topic-Sensitive PageRank はクエリ従属なランキングアルゴリズムであるため、クエリ応答時間は長くなる。

### 4.1.2 知的サーファーモデル

4.1.1 節で紹介した検索結果の改良手法は、3.2 節のランダムサーファーが挙動 3：「現在のノード内のリンクと関係なく、ランダムに選んだノードに遷移する」を選択した場合の遷移確率を、ユーザの興味・関心に合わせて変化させることで検索結果の質を向上させる手法であった。以下では、ランダムサーファーが挙動 2：「現在のノード内のアウトリン

[*47] Open Directory Project。巨大なディレクトリ型検索サイト。http://www.dmoz.org/

[*48] ODP の第 1 階層のカテゴリでは、Arts、Business、Computers、Games、Health、Home、Kids and Teens、News、Recreation、Reference、Regional、Science、Shopping、Society、Sports、World の 16 トピックに分類されている。

[*49] スタンフォード大学による Web ページのデータベース。データベースは Web 上で公開されており、ダウンロードおよび利用が可能。スタンフォード大学は Web グラフと関連する研究などでの利用を推奨している。PageRank に関連する研究の中には、WebBase を利用して実験を行っているものもある。The Stanford WebBase Project： http://www-diglib.stanford.edu/ testbed/doc2/WebBase/

クを辿って遷移する」を選択した場合の遷移確率を成分とするハイパーリンク行列 **H** を修正することで、検索結果の質を向上させる手法を紹介する [50][2][35]。

行列 **H** の成分をユーザの興味・関心に合わせて変化させる場合、ランダムサーファーの挙動 2 は挙動 2'：「現在のノード内のアウトリンクのうち、サーファーが興味・関心のあるノードへのアウトリンクを優先的に選択して遷移する」と書き換えることができる。挙動 2 ではなく挙動 2' を行うランダムサーファーを知的サーファー（intelligent surfer）とよぶ。

ランダムサーファーモデルではハイパーリンク行列 **H** の非ダングリングノードの行の成分を全て $h_{uv} = \frac{1}{n_u}$ としていたが、知的サーファーモデルの場合は行列 **H** の成分をサーファーの興味・関心の有無により変化させる。その際、行列 **H** の行の成分の合計が 1 になるように成分を変化させる。

3.3 節の Web グラフ（図）を例に、知的サーファーモデルでのハイパーリンク行列の例を挙げる [35]。Web ページ 1 には、Web ページ 2 および 3 へのアウトリンクがあるが、ユーザの遷移の記録を何らかの形で参照[*50]した結果、Web ページ 1 から Web ページ 2 への遷移は Web ページ 1 から Web ページ 3 への遷移よりも 2 倍多いことがわかったとする。そのため、ランダムサーファーモデルのハイパーリンク行列 **H** の 1 の行を次のように書き換え、知的サーファーモデルの場合のハイパーリンク行列 $\mathbf{H}'$ を作る。

$$\mathbf{H} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{4.3}$$

を次のように変更する。

$$\mathbf{H}' = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 2/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{4.4}$$

[*50] Web ページのアクセスログを取得したり、インターネットブラウザでユーザの遷移の情報を記憶・集計するなどの例が考えられる。

Baeza-Yates らは、ハイパーリンク行列 **H** の非ダングリングノードの行の成分の重み付けを行うにあたって、Web 管理者がアウトリンクに付加した情報を利用している [2]。Baeza-Yates らは Web 管理者がアウトリンクに付加した情報として以下の 3 つを評価している。

(1) HTML コード中で先に書いてあるアウトリンクほど評価を高める。
(2) アウトリンクのアンカーテキストが <h1><h2><b> などのタグで囲まれている場合、評価を高める。
(3) アウトリンクのアンカーテキストが長い場合、評価を高める。

Baeza-Yates らは各項目について次のような説明を行っている [2]。(1) については、実際にインターネットブラウザで表示されている場所が上であるか下であるかではなく、HTML コード内で先に書かれたアウトリンクを評価している[*51]。(2) については、<h1> を最も重要なタグとし、<h2> を次に重要なタグとしている。また、 や <b> についてもアウトリンクを強調するタグとして考慮している[*52]。(3) については、短いアンカーテキストの例として”home”や”here”を挙げ、そのようなアンカーテキストにはリンク先のページについての情報があまり付加されていないと述べている。一方、長いアンカーテキストが使われているアウトリンクはリンク先のページについて詳しく説明してある場合が多く、重要なアウトリンクであると考えている。

また、Richardson らは、アウトリンクの重み付けに用いる情報として、Web ページの内容とリンク先の Web ページとの関連度および、サーファーが検索したクエリの内容を考慮している [50]。この場合、検索結果の質は向上するが、クエリ従属な PageRank であるためクエリ応答時間が長くなってしまい、実用上は問題があると述べている。

知的サーファーモデルに関連する研究に共通する課題としては、記憶領域の問題が挙げられる。ランダムサーファーモデルの行列 **H** では非ダングリングノードの成分が一様で

[*51] ブラウザでの表示には HTML 以外の要素（例 CSS：Cascading Style Sheets）が関連してくる場合がある。そのため、ブラウザでの表示位置を判定する場合には複雑な処理を行う必要がある。また、表示の具合は使用するブラウザによっても異なる場合がある。HTML コード内で先に書かれたアウトリンクの評価を高くする理由としては、実装が容易であることが考えられる。

[*52] <h1><h2> タグで囲まれたテキストは、デフォルトの設定ではフォントサイズも大きく行間も大きいため、ユーザの目に留まりやすい。そのため、ユーザがこれらのタグで囲まれたアウトリンクをクリックして遷移する可能性は比較的高くなるであろう。また、 や <b> などで Web 管理者が強調したアウトリンクのリンク先の Web ページは、その Web ページのトピックと関連のある Web ページである可能性が比較的高いであろう。そのため、あるトピックに興味のあるユーザはこれらのタグで囲まれた Web ページへ遷移する可能性が比較的高いと考えられる。すなわち、これらのタグはアウトリンクに重みをつける場合に考慮すべき重要な要素であると考えられる。

あったが、知的サーファーモデルの場合には一様ではなく、それぞれの成分を個別に記憶する必要がある。また、成分の重みを決定するために必要な情報も記憶しておく必要がある。クエリ従属なランキングアルゴリズムを用いる場合には、クエリ応答時間の短縮も課題として挙げられる。

### 4.1.3 damping factor によるリンクスパムの特定

damping factor $d$ は、ランダムサーファーが現在のノード内のアウトリンクを辿って遷移する割合と、ランダムに選んだノードに遷移する割合とを決定する変数である。$d$ の値の設定は、PageRank 値に与える影響は PageRank に関連する研究において注目されている [36][57][54][1][35]。

$d$ の値の設定により影響を受ける要素として、PageRank 値を算出するべき乗法が収束するまでの反復回数、および PageRank 値自身がある。以下では、$d$ の値の設定により両者がどのような影響を受けるかを紹介するとともに、$d$ の値による影響をふまえて検索結果の質を向上させる提案について紹介する。

Meyer らは、$d$ の値を変更して PageRank 値の計算をべき乗法で行った場合、収束判定基準が $10^{-10}$ になるまでに必要な反復回数は表 2 のように変化したと述べている [35][*53]。Meyer らは、Google のインデックス数[*54]の規模でべき乗法を行った場合に反復一回あた

表 2　damping factor の値による収束判定基準までの反復回数の変化 ([35] より引用)

| $d$ | 反復回数 (回) |
|---|---|
| 0.5 | 34 |
| 0.75 | 81 |
| 0.8 | 104 |
| 0.85 | 142 |
| 0.9 | 219 |
| 0.95 | 449 |
| 0.99 | 2,292 |
| 0.999 | 23,015 |

[*53] [35] では PageRank 値の計算に用いたデータセットは明らかにしていない。

[*54] Google は Web ページのインデックス数を公開していないが、2009 年 1 月時点での Google のインデックス数は約 300 億〜400 億であると考えられる。以下にその根拠を述べる。

りに要する時間は数十分〜数時間であろうと述べている（2004 年時点：[36]）。それを踏まえて Meyer らは、Page らの $d = 0.85$ という選択について、現実的に計算を行う上で妥当な数字であると述べている [36][35]。

Haveliwala と Kamvar は、行列 $\mathbf{G}$ の固有値のうち絶対値が 2 番目に大きい固有値（準優固有値とよぶ）と、damping factor $d$ との間には以下の定理が成り立つことを証明した [27]。

**定理 1** $\mathbf{G} = d\mathbf{S} + (1-d)\,\mathbf{1}\,{}^t\mathbf{v}$ であれば、準優固有値 $\lambda_2$ は $|\lambda_2| \geq d$ である。

**定理 2** $\mathbf{G}$ の全ての部分行列のうち、既約なものが 2 つ以上存在すれば、$|\lambda_2| = d$ である。

証明は省略する [27]。

べき乗法では $\left|\frac{\lambda_2}{\lambda_1}\right|$ の値が小さいほど速く収束する。上述 [定理 2] の場合は $|\lambda_2| = d$ であるから、$d$ の値が大きければ大きいほど $\left|\frac{\lambda_2}{\lambda_1}\right|$ の値が大きくなる。また、$d$ の値が小さければ小さいほど $\left|\frac{\lambda_2}{\lambda_1}\right|$ の値が小さくなる。そのため、$d$ の値によって収束までの反復回数は表 2 のように変化する [35]。

Thorson は、$d$ の値によって反復回数が変化することに加え、$d$ の値によって異なる PageRank 値が算出されることを実験で示し、$d$ の値の変化が Web ページの順位付けおよび検索結果の質に影響する可能性があることを示した [54]。

また Zhang らは、$d$ の値によって PageRank 値が影響を受けることを活用して、リンクスパムにより意図的かつ不正に PageRank 値を高めようとする Web ページを特定し、それらの Web ページの PageRank 値にペナルティを与える手法を提案している [57]。Zhang らの研究は、検索結果の質の向上につながる可能性がある。

### 4.1.4 バックボタンモデル

以下では、バックボタンモデルに関連する研究 [17][40][52][35] を紹介する。

通常の PageRank のランダムサーファーモデルでは、サーファーはダングリングノー

---

Google は 2008 年、米国時間 7 月 25 日の公式ブログ (http://googleblog.blogspot.com/) 上で、Google が Web グラフ上に把握しているノード数が 1 兆を超えたと発表した。しかし、Google は 1 兆のノード全てをインデックスしているわけではないと述べている。

2003 年の時点で Google が発表したインデックス数は 42 億であり、2004 年の時点では 80 億であった。また、2008 年の 7 月 29 日に Google と IBM の元社員が立ち上げた検索エンジン「Cuil」(http://www.cuil.com/) の 2009 年 1 月 18 日時点でのインデックス数は 124,426,951,803（約 1244 億）であり、cuil ホームページ上に”Cuil searches more pages on the Web than anyone else - three times as many as Google and ten times as many as Microsoft. ”と述べている。このことから、2009 年 1 月時点での Google のインデックス数は約 300 億〜400 億程度であると考えられる。

ドに遷移した場合、全ての Web ページからランダムに選んだ Web ページに遷移していた。しかし、現実のサーファーがダングリングノードに遷移した場合には、ブラウザの「戻る」ボタンで 1 つ前に閲覧していた Web ページに戻る可能性がある。

「戻る」ボタンで以前閲覧していた Web ページに再び遷移する場合も考慮して PageRank 値を算出するモデルのことをバックボタンモデルとよぶ。以下、通常のランダムサーファーモデルでの挙動 2、挙動 3 をまとめて順遷移 (forward step) とよび、「戻る」ボタンで遷移する場合を逆遷移（backward step）とよぶ。

Fagin らは、マルコフ連鎖に修正を加えることでバックボタンモデルを考えている [17]。以下でその考え方を紹介する。有限状態空間を $S = \{S_1, S_2, \ldots, S_n\}$ とし、時点を表すパラメータを $t$ $(t = 0, 1, 2, \ldots)$ とする。また各状態に遷移した場合に逆遷移を行う確率を成分とするベクトルを逆遷移ベクトルとよび、$\boldsymbol{\alpha}$ で表す。時点 t における状態を $X_t \in S$ と表し、過去に遷移した状態を記憶するスタック[*55]を履歴スタックとよび $H$ で表す。時点 $t$ での履歴スタックを $H_t$ とあらわし、$\mathrm{top}(H)$ でスタック $H$ の先頭に記憶されている状態を表す $(\mathrm{top} : \mathrm{stack} \to S)$。時点 $t = 0$ では状態は $X_0 \in S$ であり、そのときのスタック $H_0$ を $H_0 = [X_0]$ とする。時点を $t$ として、順遷移および逆遷移のしかたは以下の規則に従う。

(1) $H_t = [X_0]$ であれば、順遷移を行う。
(2) (1) でなければ、確率 $\alpha_{\mathrm{top}(H_t)}$ で逆遷移を行い、確率 $1 - \alpha_{\mathrm{top}(H_t)}$ で順遷移を行う。

(2) の場合、順遷移および逆遷移を以下のように行う。
（ただし $\mathrm{push} : \mathrm{stack} \times S \to \mathrm{stack},\ \mathrm{pop} : \mathrm{stack} \to \mathrm{stack}$ とする。）

- 順遷移の場合、状態 $X_t$ における遷移確率に従って状態 $X_{t+1}$ へ遷移する。遷移後、$X_{t+1}$ を履歴スタック $H_t$ に push して記憶する。時点 $t+1$ での履歴スタックは $H_{t+1} = \mathrm{push}(H_t, X_{t+1})$ となる。
- 逆遷移の場合、$H_{t+1} = \mathrm{pop}(H_t)$ とし、$\mathrm{top}(H_{t+1})$ へ遷移する（遷移先の状態 $X_{t+1}$ は $\mathrm{top}(H_{t+1})$ と同じになる）。

例えば時点 $t$ でのスタックが $H_t = [X_{i_1}, \ldots, X_{i_n}]$ であり、時点 $t+1$ で $X_{t+1}$ へ順遷移する場合には、時点 $t+1$ でのスタックは $H_{t+1} = [X_{t+1}, X_{i_1}, \ldots, X_{i_n}]$ となる。また、時点 $t$ でのスタックが $H_t = [X_{i_1}, X_{i_2}, \ldots, X_{i_n}]$ であり、時点 $t+1$ で $X_{t+1} = X_{i_2}$ へ逆遷移

[*55] プッシュダウンスタック (pushdown stack)。後入れ先出し（Last In First Out）でデータを保持するデータ構造。スタックにデータを挿入する操作をプッシュ（push）、データを取り出す操作をポップ（pop）という [59]。

する場合には、時点 $t+1$ でのスタックは $H_{t+1} = [X_{i_2}, \ldots, X_{i_n}]$ となる。順遷移・逆遷移いずれを行った場合でも、一連の操作の後には常に $\mathrm{top}(H_{t+1}) = X_{t+1}$ である。Fagin らは [17] で、この設定の下では常に定常分布をもつことを示している。また、その定常分布は通常のマルコフ連鎖とは異なり初期状態によって変化することを示している。

Sydow は、Fagin らの [17] をもとにバックボタンモデルに別の設定を行っている [52]。Sydow のバックボタンモデルでは、ランダムサーファーが確率 $a$ $(0 \leq a < 1)$ で現在のノード内のアウトリンクを辿って遷移し、確率 $b$ $(0 \leq b < 1)$ で逆遷移を行い、確率 $1-(a+b)$ でランダムに選んだノードに遷移する場合を考えている。また、逆遷移も 1 つのアウトリンクとして PageRank 値を与えるものとしている。Sydow は、このモデルでは PageRank 値の計算に必要なメモリが通常の PageRank 値の計算の 2 倍以上になる [52] と述べている。一方で、反復計算に要する時間は通常の PageRank と変わらないと述べ、検索結果の質の向上につながる可能性があると述べている。

Meyer らは、Fagin らや Sydow のバックボタンモデルは複雑であり、通常のマルコフ連鎖の枠も超えてしまうと述べ、よりシンプルなバックボタンモデルを提案している [35]。Meyer らは、ランダムサーファーがダングリングノードに遷移した場合の挙動 1 を変更した挙動 1'：ダングリングノードに遷移した場合には次の遷移で逆遷移するという挙動を行うランダムサーファーを提案している。このバックボタンモデルをバウンスバック（bounce-back）モデルとよぶ。以下で、バウンスバックモデル [35] について説明する。

Meyer らは、逆遷移をアウトリンクとして解釈するため、ダングリングノードがもつそれぞれのインリンクに対して、バウンスバックノードを作成している。バウンスバックノードとは、ダングリングノードへアウトリンクをしているノードにだけアウトリンクをもつノードのことである。バウンスバックノードはダングリングノードがもつインリンクの数だけ作成される。

バウンスバックモデルではハイパーリンク行列 $\bar{\mathbf{H}}$ のサイズは非常に大きくなってしまう。その一方で、ハイパーリンク行列 $\bar{\mathbf{H}}$ は全ての行の成分の合計が 1 である確率行列であるから stochasticity adjustment（3.7 節）を行う必要はない。そのため、ハイパーリンク行列 $\bar{\mathbf{H}}$ に対しては primirivity adjustment のみを行う。式（4.1）の行列 $\mathbf{G}$ をバウンスバックモデルに書き換えると、その行列 $\bar{\mathbf{G}}$ は以下のように表される。

$$\bar{\mathbf{G}} = d\bar{\mathbf{H}} + (1-d)\,{}^t\mathbf{v} \tag{4.5}$$

以下で、バウンスバックモデルでの Web グラフの変化の例を述べる。図 7 の例では行

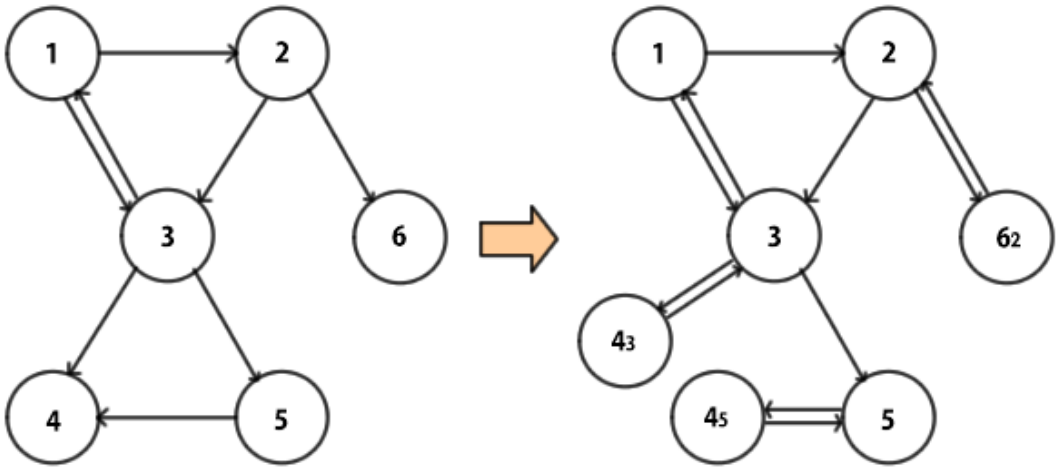


図 7 （左）通常の Web グラフの例 （右）バウンスバックモデルの Web グラフの例

列 $\mathbf{H}$ は以下のように表される。

$$\mathbf{H} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{4.6}$$

またバウンスバックモデルでのハイパーリンク行列 $\bar{\mathbf{H}}$ は以下のように表される。

$$\bar{\mathbf{H}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4_3 \\ 4_5 \\ 5 \\ 6_2 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4_3 & 4_5 & 5 & 6_2 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{4.7}$$

このように、行列 $\bar{\mathbf{H}}$ は通常の行列 $\mathbf{H}$ に比べて

（バウンスバックノードの数）−（ダングリングノードの数)

の個数分だけ成分が多くなる。

バウンスバックモデルでの PageRank 値の計算は以下の手順で行う [35]。

(1) バウンスバックノードを含めた状態で全ての Web ページの PageRank 値を計算する。
(2) 各ダングリングノードに対応するバウンスバックノードの PageRank 値を合計する。
(3) (2) で求めた値をダングリングノードの PageRank 値とする。

この手順で、もとの Web グラフにおける全ての Web ページの PageRank 値を算出する。

Meyer らは、6 つのノードのうち 2 つのノードがダングリングノードの小さな Web グラフで上記の手順により PageRank 値を求めた結果、通常の PageRank 値とバウンスバックモデルの PageRank 値には差が見られたと述べている [35]。

## 4.2 連立一次方程式による PageRank 値の計算

Page らは PageRank 値の算出にあたって、行列 $\mathbf{G}$ の最大固有値 1 に対応する固有ベクトルである PageRank ベクトルをべき乗法により算出した [48]。べき乗法による PageRank 値の算出にはさまざまな利点がある (3.7 節)。べき乗法は現在も主要な PageRank 値の計算方法である。

べき乗法以外の PageRank ベクトルの計算方法としては、連立一次方程式（線形方程式系）による PageRank ベクトルの計算がある。連立一次方程式による PageRank ベクトルの計算は改良研究でしばしば用いられている。本小節では連立一次方程式を用いた PageRank ベクトルの計算について紹介する [35]。

べき乗法による PageRank 値の計算式 (3.12) は、次の式で表された。

$$ {}^t\mathbf{r} = {}^t\mathbf{r}\mathbf{G} \quad \text{かつ} \quad {}^t\mathbf{r}\mathbf{1} = 1 \tag{4.8} $$

式 (4.8) から、連立一次方程式の式を導出する [35]。

$$ {}^t\mathbf{r}\,(\mathbf{I} - \mathbf{G}) = {}^t\mathbf{0} \quad \text{かつ} \quad {}^t\mathbf{r}\mathbf{1} = 1 \tag{4.9} $$

式 (4.9) に p.32 の式 (3.9) を代入し、次のように変形する。（ただし、ランクソースベクトル ${}^t\mathbf{e} = \frac{1}{n}\,{}^t\mathbf{1}$ ではなくパーソナライズドランクソースベクトル ${}^t\mathbf{v}$ を用いる[*56]。）

$$ {}^t\mathbf{r}\left(\mathbf{I} - \left(d\mathbf{S} + (1-d)\,\mathbf{1}\,{}^t\mathbf{v}\right)\right) = {}^t\mathbf{0} \tag{4.10} $$

よって、式 (4.9) は

$$ {}^t\mathbf{r}\,(\mathbf{I} - d\mathbf{S}) = (1-d)\,{}^t\mathbf{v} \tag{4.11} $$

と表すことができる。式 (4.11) の $(\mathbf{I} - d\mathbf{S})$ には次のような性質がある（証明は [4][18][42]）。

1. 行列 $(\mathbf{I} - d\mathbf{S})$ は Minkowski 行列[*57]である。
2. 行列 $(\mathbf{I} - d\mathbf{S})$ は正則行列である。
3. 行列 $(\mathbf{I} - d\mathbf{S})$ の行の合計は $1 - d$ である[*58]。

[*56] パーソナライズドランクソースベクトル：4.1.1 節

[*57] 以下が成り立つ場合、行列 $\mathbf{A}$ を Minkowski 行列という。
「$\mathbf{A} = r\mathbf{I} - \mathbf{B}$ を満たす $\mathbf{B} \geq \mathbf{0}$ および実数 $r \geq \rho(\mathbf{B})$ が存在する。」

[*58] ただし、非ダングリングノードが 1 つも存在しないときは成り立たない

4. $||\,(\mathbf{I}-d\mathbf{S})\,||_\infty = 1+d$ である。
5. $(\mathbf{I}-d\mathbf{S})$ は Minkowski 行列であるから、$(\mathbf{I}-d\mathbf{S})^{-1} \geq 0$ である。
6. $(\mathbf{I}-d\mathbf{S})^{-1}$ の行の合計は $\frac{1}{(1-d)}$ である。そのため、$||\,(\mathbf{I}-d\mathbf{S})^{-1}\,||_\infty = (1-d)^{-1}$ である。
7. 条件数[*59] $\kappa_\infty\,(\mathbf{I}-d\mathbf{S}) = \frac{(1+d)}{(1-d)}$ である。

特に性質 2 は重要である。

式（4.11）でも解くことができるが、行列 $\mathbf{S}$ のダングリングノードの行は密であり、計算に必要な記憶領域や処理量も膨大になる。そのため式（4.11）に p.31 の式 (3.8) を代入し、行列 $\mathbf{H}$ の式を導出する [35]。

$$
\begin{aligned}
{}^t\mathbf{r}\left(\mathbf{I}-d\left(\mathbf{H}+\mathbf{a}\,{}^t\mathbf{v}\right)\right) &= (1-d)^t\mathbf{v} \\
{}^t\mathbf{r}(\mathbf{I}-d\mathbf{H}) &= \left(1-d+d\,{}^t\mathbf{r}\mathbf{a}\right)\,{}^t\mathbf{v}
\end{aligned}
\tag{4.12}
$$

式 (4.12) の $\mathbf{a}$ はダングリングノードベクトルであり、${}^t\mathbf{r}\mathbf{a}$ はダングリングノードの PageRank 値の合計値である。ダングリングノードの PageRank 値の合計を $\gamma$ とおき、式 (4.12) を次のように変形する。

$$
{}^t\mathbf{r}(\mathbf{I}-d\mathbf{H}) = (1-d+d\gamma)\,{}^t\mathbf{v} \tag{4.13}
$$

ここで、ダングリングノードの PageRank 値の合計を $\gamma = 1$ であると仮定する。すると、全ての Web ページの PageRank 値の合計 $||\mathbf{r}||_1 \neq 1$ となってしまう。しかしながら、PageRank 値はその相対的な大きさによって Web ページの順位付けを行うための値であるから、$\gamma = 1$ という仮定の下で全ての Web ページの仮の PageRank ベクトル $\mathbf{x}$ を求めた後で、$\mathbf{x}$ の成分の合計値で $\mathbf{x}$ の各成分を割ることで全ての Web ページの PageRank 値の合計が 1 である真の PageRank ベクトル $\mathbf{r}$ を求めても、Web ページの順位付けを行う用途には影響しない。

$\gamma = 1$ を仮定した場合の、仮の PageRank ベクトルを $\mathbf{x}$ と表す（$||\mathbf{x}||_1 \neq 1$）。この条件の下で式 (4.13) を表すとともに、$||\mathbf{r}||_1 = 1$ である真の PageRank ベクトルを求めるために、$\mathbf{x}$ の全ての成分の合計値で $\mathbf{x}$ の全ての成分を除する。

$$
{}^t\mathbf{x}(\mathbf{I}-d\mathbf{H}) = {}^t\mathbf{v} \quad \text{かつ} \quad {}^t\mathbf{r} = \frac{{}^t\mathbf{x}}{{}^t\mathbf{x}\,\mathbf{1}} \tag{4.14}
$$

式（4.14）が疎な行列 $\mathbf{H}$ を用いた連立一次方程式による PageRank 値の計算式である。式（4.14）により PageRank ベクトルを正しく求めることができることは Meyer らが証明している（[35] §7.3)。

[*59] condition number $\kappa_p = ||\mathbf{A}||_p\,||\mathbf{A}^{-1}||_p$

式 (4.14) の $(\mathbf{I}-d\mathbf{H})$ は、$(\mathbf{I}-d\mathbf{S})$ の性質の多くを引き継いでいる（証明は [4][18][42]）。

1. 行列 $(\mathbf{I}-d\mathbf{H})$ は Minkowski 行列である。
2. 行列 $(\mathbf{I}-d\mathbf{H})$ は正則行列である。
3. 行列 $(\mathbf{I}-d\mathbf{H})$ の行の合計は、非ダングリングノードの行であれば $1-d$ で、ダングリングノードの行であれば 1 である。
4. $||(\mathbf{I}-d\mathbf{H})||_\infty = 1+d$ である[*60]。
5. $(\mathbf{I}-d\mathbf{H})$ は Minkowski 行列であるから、$(\mathbf{I}-d\mathbf{H})^{-1} \geq 0$ である。
6. $(\mathbf{I}-d\mathbf{H})^{-1}$ の行の合計は非ダングリングノードの行であれば $\frac{1}{(1-d)}$ 以下であり、ダングリングノードの行であれば 1 である。
7. 条件数 $\kappa_\infty(\mathbf{I}-d\mathbf{H}) \leq \frac{(1+d)}{(1-d)}$ である。
8. $(\mathbf{I}-d\mathbf{H})^{-1}$ の行のうち、ダングリングノード $i$ による行は ${}^t\mathbf{e}_i$ である ( ${}^t\mathbf{e}_i$ は単位行列の $i$ 番目の行を表す)。

特に性質 2 は重要である[*61]。

連立一次方程式による PageRank 値の計算には、次の利点がある [35]。

- 企業内の Web ページなど、Web ページの総数が少ない場合には、べき乗法よりも計算を速く行うことができる。
- damping factor が 1 に近づくにつれて反復回数が大きくなってしまう問題 (4.1.3) は、連立一次方程式の場合は起こらない（ただし、PageRank 値はべき乗法と同様に damping factor の影響を受ける。）。
- PageRank 値の計算の主流であるべき乗法に比べて、研究の余地が大きい。

[*60] ただし、非ダングリングノードが 1 つも存在しないときは成り立たない

[*61] 例えば、4.4.1 節で活躍する。

## 4.3 PageRank 値の計算にかかる記憶領域の節約

本小節では、PageRank 値の計算にかかる記憶領域の節約に関連する研究を紹介する。PageRank 値を計算するためには、PageRank 値の計算に必要な要素を記憶媒体に記憶しておく必要がある。その際、記憶する情報を少なくすることができれば、PageRank 値の計算の際に読み込む情報を少なくすることができ、計算にかかる処理量を軽減できる可能性がある。また、記憶してある情報を高速に読み書きできれば、PageRank 値の計算にかかる時間の短縮につながる可能性がある。

Meyer らの PageRank 値の定義の場合、PageRank 値の計算に必要な要素としてはハイパーリンク行列 $\mathbf{H}$、ダングリングノードベクトル $\mathbf{a}$、パーソナライズドランクソースベクトル ${}^t\mathbf{v}$[*62]、$k$ 回目の反復時の PageRank ベクトル ${}^t\mathbf{r}$ がある [35] 。各要素の記憶に必要な記憶領域は表 3 の通りである[*63]。ただし、$nnz(\mathbf{H})$ は行列 $\mathbf{H}$ 内の非ゼロ要素の数を

表 3　PageRank 値の計算に必要な要素とその記憶領域 （[35]8 章より。ただし $n$ および「必要な記憶領域」は本稿で挿入し、$\mathbf{a}$ のデータ型を int 型から long int 型に変更した。）

| 記憶する要素 | 記憶するデータ | 必要な記憶領域 |
|---|---|---|
| $\mathbf{H}$ | $nnz(\mathbf{H})$ 個の double 型のデータ | $nnz(\mathbf{H}) \times 8$ バイト以上 |
| $\mathbf{a}$ | $\lvert D\rvert$ 個の long int 型のデータ | $\lvert D\rvert \times 8$ バイト |
| ${}^t\mathbf{v}$ | $n$ 個の double 型のデータ | $n \times 8$ バイト |
| ${}^t\mathbf{r}^{(k)}$ | $n$ 個の double 型のデータ | $n \times 8$ バイト |
| $n$ | $n$ 個の double 型のデータ | $n \times 8$ バイト |

表し、$\lvert D\rvert$ はダングリングノードの数を表す。このうち、最も記憶領域を必要とする要素は行列 $\mathbf{H}$ である。そのため、記憶領域の問題では行列 $\mathbf{H}$ をいかに効率よく記憶するかが課題となる。

例えば、2009 年 1 月時点の Google のインデックス数が 300 億であるとする。ま

[*62] ただし、ランクソースベクトルとして ${}^t\mathbf{e} = \dfrac{{}^t\mathbf{1}}{n}$ を用いる場合は記憶する必要はない。

[*63] Meyer らは、表 3 では行列 $\mathbf{H}$ の記憶に必要なデータは $nnz(\mathbf{H})$ 個の double 型であると述べている。これらのデータを記憶する際には、行列 $\mathbf{H}$ 内のどの位置にデータが存在するかを記憶するための領域が必要である。

た、1 つの Web ページに平均 10 本のアウトリンクが存在するとすると[*64]$nnz(\mathbf{H}) = 300,000,000,000$ 個の double 型のデータを記憶する必要がある。つまり、2.4TB（テラバイト）の記憶領域が必要となる。この場合、PageRank 値の計算を行う計算機のメモリ（主記憶装置）上に行列 **H** の情報を記憶することができない場合も考えられる。

PageRank 値の計算に際して、メモリ上に行列 **H** の情報を記憶することができない場合には、行列 **H** のデータを圧縮してメモリ上に記憶できるような状態にするか、行列 **H** のデータの圧縮を行わずにハードディスク（補助記憶装置）に記憶する必要がある。後者の場合には、ハードディスクの読み書きの効率を上げる必要がある。

### 4.3.1 隣接リストによるハイパーリンク行列の圧縮

記憶領域を節約するための手法としては、ハイパーリンク行列 **H** の情報を隣接リストに表現して記憶する手法がある。隣接リストとは、Web グラフ上のノードに隣接する（リンクでつながっている）ノードをリストにして表したものである。隣接リストには、ある Web ページがもつアウトリンクとインリンクとのいずれか、もしくは両方を記憶する。

隣接リストの記憶に必要な記憶領域を節約するために、隣接リストのデータを圧縮する手法が研究されている。その手法として、本稿では Bharat らによる gap technique[5] および Raghavan らによる reference encoding technique[49] を紹介する。

Bharat らの gap technique[5] では、Web グラフ上に存在するハイパーリンクの局所性を利用して隣接リストを圧縮する。Web グラフ上のハイパーリンクの局所性とは、リンク元の Web ページとリンク先の Web ページとに割り振られたラベル（例えば、docID）の数字が近くなることが多い、という性質のことである。

例えば、ある Web ページにつけられた docID が 300 であったとする（以下、この小節に限り Web ページを docID でよぶ。）。ハイパーリンクの局所性がある場合、300 は、306、307、309、313 など docID が近い Web ページからのインリンクをもつことが多い。一方、300 は docID が離れている Web ページ（29,316 や 456,068,393 など）からのインリンクをもつことは少ない。その場合、gap technique では 300 のインリンクの情報を表 4 のように記憶する [5]。記憶すべき数字は 300、306、0、1、3 である。gap method では、そのラベルと次のラベルの間の隔たり（gap）を記憶する。表 4 の 306 の次の 0 は、306 と 307 との docID の間の隔たり[*65]がいくつであったかを示す数字である。307 と 309 と

[*64] Meyer らは 10 本という数字を使っているが、1 つの Web ページに平均何本のアウトリンクが存在するかはデータセットにより異なる。例えば、Kamvar らは 1 つの Web ページあたり 8 本のアウトリンクが存在したと述べており [32]、Kleinberg らは 7.2 本のアウトリンクが存在したと述べている [33]。

[*65] 二つの Web ページの docID の差に 1 を引いた数である。

表 4 隣接リストで 300 がもつ 306 307 309 313 からのインリンクを記憶した例

| Web ページの docID | docID へのリンク元の Web ページ |
|---|---|
| ⋮ | ⋮ |
| 300 | 306 0 1 3 |
| ⋮ | ⋮ |

の docID の間の隔たりが 1 であるため、1 を記憶する。また、309 と 313 との docID の間の隔たりは 3 であるため、3 を記憶する。このように記憶することで、307 や 309、313 という 3 桁の数字を記憶する必要がなくなる。

docID は数十億から数百億に上る可能性がある。また、1 つの Web ページがもつインリンクの数が数百・数千・数万・数十万に及ぶ場合もある。そのような場合、ある Web ページがもつインリンクのリンク元 Web ページの docID をすべてそのまま記憶するには膨大な記憶容量が必要である。Web グラフのハイパーリンクに局所性がある場合は、扱う Web ページ数およびインリンクの数が膨大になればなるほど、gap technique により節約できる記憶領域は大きくなる。

隣接リストによるハイパーリンク行列 **H** を圧縮する別の手法としては、Raghavan らによる reference encoding technique[49] がある。reference encoding technique では、Web ページ間の類似性 (similarity) を利用する。

Web ページ間に類似性があるとは、Web ページ間で扱っているトピックが同じであったり、Web ページ間に局所性がある場合をいう。Web ページ間に類似性がある場合、それらの Web ページのアウトリンクには共通するものが多い場合がある。その場合、類似性のある Web ページのアウトリンクを記憶した隣接リストには、共通するリンク先 docID が多く含まれる。

Raghavan らは reference encoding technique の例として次の例を提示している [49]。類似性がある 2 つの Web ページを $i, j$ と表し、Web ページ $i$ の隣接リストの中に、Web ページ $j$ の隣接リストが多く含まれているとする。その場合、$i$ を $j$ に対する reference page とよぶ。

- $i = (5 \quad 7 \quad 12 \quad 89 \quad 101 \quad 190 \quad 390)$
- $j = (5 \quad 6 \quad 12 \quad 50 \quad 101 \quad 190)$

$j$ の隣接リストは、$i$ の隣接リストと同じ長さの共有ベクトル（sharing vector）と非類似

ベクトル (dissimilarity vector) とで表すことができる。共有ベクトルは reference page の隣接リストと同じ大きさのバイナリベクトル（成分が 0 と 1 のみのベクトル）である。$i$ の隣接リストの $n$ 番目（$n = 1, 2, \ldots$）の要素が $j$ の隣接リストに現れれば、共有ベクトルの $n$ 番目の成分を 1 とし、そうでなければ 0 とする。また、$j$ の要素のうち reference page $i$ の隣接リストに出現しない要素を非類似ベクトルの成分とする。上記の例の $j$ は、共有ベクトルと非類似ベクトルを使って次のように表される。

- $j = (1010110)\ (6 \quad 50)$

良い reference page を発見することができれば、バイナリの共有ベクトルで $j$ の隣接リストの大部分を記憶することができるため、大幅に記憶領域を減らすことができる。

reference encoding technique では、Web ページ間のアウトリンクのリストの類似度が高ければ高いほど多くの記憶領域を節約することができる。そのためには、良い reference page を選択しなければならない。Raghavan らは、reference page をどのように決定すべきかという考え方を提示している [49]。

## 4.4 PageRank 値の計算の高速化

本小節では、PageRank 値の計算を高速化する手法を紹介する。

PageRank 値の計算の高速化は重要な問題である。例えば、4.1.1 節の Topic-sensitive PageRank では、16 個の PageRank ベクトルを計算しなければならない。数十億の Web ページの PageRank 値を通常のべき乗法で算出する場合、1 個の PageRank ベクトルを求めるのに数日を要する [32] ことは、パーソナライズド検索を実現するための障害となる。また、Web ページは日々増え続けているため、PageRank 値の計算の規模は日々大きくなっていく。PageRank 値の計算の高速化のニーズは強いであろう。

### 4.4.1 ダングリングノードの状態集約

Brin と Page はダングリングリンクを除外して PageRank 値を計算していた [48]。ダングリングリンクのリンク先であるダングリングノードの中には、非ダングリングノードと同様に、多くのインリンク（ダングリングリンク）をもつ重要な Web ページが存在する。本節では、ダングリングリンクを除外するのではなく、ダングリングノードを 1 つの状態に集約して（まとめて）PageRank 値の計算を行う手法を紹介する [39][16][37]。

Lee らは、行列 $\mathbf{H}$（および $\mathbf{S}$、$\mathbf{G}$）のダングリングノードの行は全て同じ成分であることに着目した [39]。まず、行列 $\mathbf{H}$（および $\mathbf{S}$、$\mathbf{G}$）の行を非ダングリングノードの行とダングリングノードの行とに並び替える。次に、ダングリングノードの行を 1 つの状態：テレポーテーション状態（teleportation state）にまとめた行列を作る。その行列は、$(|\mathrm{ND}|+1)\times(|\mathrm{ND}|+1)$ の正方行列となる（$|\mathrm{ND}|$ は非ダングリングノードの数）。その行列に対して連立一次方程式を用いて PageRank ベクトルを求める。

この手法ではダングリングノードを 1 つの状態にまとめて PageRank 値を計算するため、1 つ 1 つのダングリングノードの PageRank 値をどのように復元するかが問題となる。Lee らは、Aggregation/Disaggregation 法[*66]を用いた説明を行っている。

ここでは、Langville らの提案した説明を紹介する。この方法の場合にはダングリングノード全体を 1 つのノードにまとめる代わりに行列 $\mathbf{H}$ の並び替えを行う。具体的には、行列 $\mathbf{H}$ の行を非ダングリングノードの行とダングリングノードの行とに並び替えた場合、

[*66] Aggregation/Disaggregation 法とは、nearly uncoupled markov chains (nearly completely decomposable markov chains) の定常分布を効率的に算出する手法である [11][28]

行列 $\mathbf{H}$ の成分を以下のように表すことができる（以下は [35] に従って式を導いている）。

$$\mathbf{H} = \begin{matrix} \\ \text{ND} \\ \text{D} \end{matrix} \begin{matrix} \begin{matrix} \text{ND} & \text{D} \end{matrix} \\ \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ 0 & 0 \end{pmatrix} \end{matrix} \tag{4.15}$$

ただし、ND は非ダングリングノードの行、D はダングリングノードの行を表す。ベクトル ${}^t\mathbf{v}$ についても、行列 $\mathbf{H}$ と対応付けて並び替える。

$${}^t\mathbf{v} = \begin{matrix} \begin{matrix} \text{ND} & \text{D} \end{matrix} \\ \begin{bmatrix} {}^t\mathbf{v}_1 & {}^t\mathbf{v}_2 \end{bmatrix} \end{matrix} \tag{4.16}$$

式 (4.15) の行列 $\mathbf{H}$ を用いて、連立一次方程式による PageRank 値の計算式（4.14）の係数行列 $(\mathbf{I} - d\mathbf{H})$ を以下のように表すことができる。

$$(\mathbf{I} - d\mathbf{H}) = \begin{pmatrix} \mathbf{I} - d\mathbf{H}_{11} & -d\mathbf{H}_{12} \\ 0 & \mathbf{I} \end{pmatrix} \tag{4.17}$$

ただしここで $\mathbf{I}$ は 3 種類の単位行列を表している。行列 $(\mathbf{I} - d\mathbf{H})$ および $(\mathbf{I} - d\mathbf{H}_{11})$ は正則である (4.2 節) から、式 (4.17) の逆行列 $(\mathbf{I} - d\mathbf{H})^{-1}$ を以下のように表すことができる。

$$(\mathbf{I} - d\mathbf{H})^{-1} = \begin{pmatrix} (\mathbf{I} - d\mathbf{H}_{11})^{-1} & d(\mathbf{I} - d\mathbf{H}_{11})^{-1}\mathbf{H}_{12} \\ 0 & \mathbf{I} \end{pmatrix} \tag{4.18}$$

式 (4.18) を用いると、仮の PageRank ベクトル $\mathbf{x}$ についての式 (4.14) を、両辺に右から $(\mathbf{I} - d\mathbf{H})^{-1}$ を乗じて次のように表すことができる。

$$\begin{aligned} {}^t\mathbf{x} &= {}^t\mathbf{v}(\mathbf{I} - d\mathbf{H})^{-1} && (4.19) \\ &= \begin{bmatrix} {}^t\mathbf{v}_1(\mathbf{I} - d\mathbf{H}_{11})^{-1} & d\,{}^t\mathbf{v}_1(\mathbf{I} - d\mathbf{H}_{11})^{-1}\mathbf{H}_{12} + {}^t\mathbf{v}_2 \end{bmatrix} && (4.20) \end{aligned}$$

Langville らは、式 (4.19) および式 (4.20) より PageRank ベクトルを求めるアルゴリズムを提案している [35]。（ただし、仮の PageRank ベクトルを ${}^t\mathbf{x} = [{}^t\mathbf{x}_1 \quad {}^t\mathbf{x}_2]$ とする。）

1. ${}^t\mathbf{x}_1(\mathbf{I} - d\mathbf{H}_{11}) = {}^t\mathbf{v}_1$を${}^t\mathbf{x}_1$について解く。
2. ${}^t\mathbf{x}_2 = d\,{}^t\mathbf{x}_1\mathbf{H}_{12} + {}^t\mathbf{v}_2$ を計算する。
3. ${}^t\mathbf{r} = \dfrac{[{}^t\mathbf{x}_1 \quad {}^t\mathbf{x}_2]}{||\,[{}^t\mathbf{x}_1 \quad {}^t\mathbf{x}_2]\,||_1}$ を計算して真の PageRank ベクトルを求める。

ダングリングノードが全ての Web ページの 80% を占める場合、Lee らの手法 [39] を用いると、通常の PageRank 計算に比べて PageRank ベクトルの算出にかかる処理量を $\frac{1}{5}$ に減じることができ、計算時間も飛躍的に向上する。

### 4.4.2 収束判定基準の変更による効率化

べき乗法の収束判定基準を見直すことで PageRank 値の計算を高速化する手法を紹介する。べき乗法による PageRank ベクトルの計算では、$k$ 回目の反復時点の PageRank ベクトル ${}^t\mathbf{r}^{(k)}$ と $k+1$ 回目の反復時点の PageRank ベクトル ${}^t\mathbf{r}^{(k+1)}$ の成分の差の合計 $||{}^t\mathbf{r}^{(k+1)} - {}^t\mathbf{r}^{(k)}||_1$ が収束判定基準 $\epsilon$ を下回れば、PageRank ベクトルが収束した（定常状態に達した）と判定していた。

Haveliwala は、PageRank ベクトルの計算において、必ずしも真の PageRank 値を求める必要がないことに着目した [26]。PageRank 値はその相対的な大きさによって Web ページの順位付けを行うための値であるため、正確な順位付けさえ可能であれば、PageRank 値の近似値を算出できた時点で収束したと判定すればよいという考えを提示した。

Haveliwala は、収束判定基準の見直しによる PageRank 値の計算の高速化の可能性があることを実験により示している [26]。まず、べき乗法の反復を 5 回・10 回・25 回・50 回・100 回行って PageRank 値を算出し、それらの用いて Web ページの順位付けを行っている。次に、反復を 100 回行って求めた PageRank 値による順位付けと、反復を 5 回・10 回・25 回・50 回行って求めた PageRank 値の順位付けとの類似度[*67]を求めている。クエリによる差はあるが、10 回程度の反復でも 7〜8 割程度の正確な順位付けができることがグラフ [26] から読み取れる。Haveliwala はこの結果を受けて、収束判定基準の見直しにより、PageRank 値計算にかかる時間を短縮することができる可能性があると述べている。

また Meyer らは、damping factor $d$ の値の設定によって、べき乗法が収束するまでに要する反復回数が変化することから、$d$ の値を 0.85 付近ではなく例えば 0.8 以下に下げることで、収束までに必要な反復回数を減らすことができると述べている [35]。ただしその場合には、PageRank 値と Web のハイパーリンク構造の関連は低くなり、「重要な Web ページからのインリンクをもつ Web ページを重要な Web ページであると判断する」という PageRank 値の直感的な考え方 (3.1 節 p.20）には反する方向へ進む。

[*67] 2 つの PageRank 値による Web ページの順位付けでそれぞれの上位 10 個に現れたすべての Web ページのうち、共通する Web ページがいくつあったかを類似度としている。

### 4.4.3 Extrapolation

Kamvar らは、PageRank ベクトルをべき乗法で求めるにあたって、収束までに必要な反復回数を減じるアルゴリズム（Aitoken Extrapolation および Quadratic Extrapolation）を提案している [32]。これらのアルゴリズムは、マルコフ行列 $\mathbf{G}$ の定常分布すなわち PageRank ベクトル ${}^t\mathbf{r}$ を推測することで反復回数を減らす手法である。以下で、その概要を紹介する。

Aitken Extrapolation では、以下を仮定する (以下は [32] に従って式を導いている)。$k$ 回目の反復時点での PageRank ベクトル ${}^t\mathbf{r}^{(k)}$ が、行列 $\mathbf{G}$ の最大固有値 1 に対応する固有ベクトル ${}^t\mathbf{r}$ と、準優固有値 $\lambda_2$ に対応する固有ベクトル ${}^t\mathbf{s}$ との線形結合で以下のように表せると仮定する（ただし、$1 > \lambda_2 \geq \ldots \geq \lambda_n$ とする)。

$$ {}^t\mathbf{r}^{(k)} \fallingdotseq {}^t\mathbf{r} + \alpha\ {}^t\mathbf{s} \tag{4.21} $$

すると、${}^t\mathbf{r}^{(k+1)}$ および ${}^t\mathbf{r}^{(k+2)}$ はそれぞれ以下のようになる。

$$ {}^t\mathbf{r}^{(k+1)} = {}^t\mathbf{r}^{(k)}\mathbf{G} \fallingdotseq {}^t\mathbf{r} + \alpha\lambda_2\ {}^t\mathbf{s} \tag{4.22} $$

$$ {}^t\mathbf{r}^{(k+2)} = {}^t\mathbf{r}^{(k+1)}\mathbf{G} \fallingdotseq {}^t\mathbf{r} + \alpha\lambda_2^2\ {}^t\mathbf{s} \tag{4.23} $$

${}^t\mathbf{r}$ の $i$ 番目の成分を $r_i$ と表し、ベクトル ${}^t\mathbf{g}$ および ${}^t\mathbf{h}$ を以下のように定義する ($g_i$ および $h_i$ はベクトル ${}^t\mathbf{g}$ および ${}^t\mathbf{h}$ の $i$ 番目の成分を表す)。

$$ g_i = (r_i^{(k+1)} - r_i^{(k)})^2 \tag{4.24} $$

$$ h_i = r_i^{(k+2)} - 2r_i^{(k+1)} + r_i^{(k)} \tag{4.25} $$

この ${}^t\mathbf{g}$ および ${}^t\mathbf{h}$ は ${}^t\mathbf{r}^{(k)}$,${}^t\mathbf{r}^{(k+1)}$,${}^t\mathbf{r}^{(k+2)}$ から求めることができる。${}^t\mathbf{s}$ の $i$ 番目の成分を $s_i$ と表すと、式（4.22）より、

$$ g_i \fallingdotseq \alpha^2(\lambda_2 - 1)^2(s_i)^2 \tag{4.26} $$

$$ h_i \fallingdotseq \alpha(\lambda_2 - 1)^2(s_i) \tag{4.27} $$

と表すことができる。このとき、$h_i \neq 0$ であると仮定し、ベクトル ${}^t\mathbf{f}$ を以下のように定義する。

$$ f_i = \frac{g_i}{h_i} = \alpha\ s_i \tag{4.28} $$

${}^t\mathbf{f}$ は ${}^t\mathbf{r}^{(k)}$,${}^t\mathbf{r}^{(k+1)}$,${}^t\mathbf{r}^{(k+2)}$ から求めることができる。式 (4.26),（4.28）より

$$ {}^t\mathbf{f} = \alpha\ {}^t\mathbf{s} \tag{4.29} $$

であり、${}^t\mathbf{f}$ は $\lambda_2$ に対応する左固有ベクトルになる。式 (4.29) を式 (4.21) に代入することで、PageRank ベクトルを次のように推定することができる。

$$ {}^t\mathbf{r} \fallingdotseq {}^t\mathbf{r}^{(k)} - {}^t\mathbf{f} \tag{4.30}$$

このように Aitken Extrapolation は、$k$ 回目の反復時の PageRank ベクトル ${}^t\mathbf{r}^{(k)}$ から $\lambda_2$ に対応する固有ベクトル ${}^t\mathbf{f}$ を引くことで、真の PageRank ベクトル ${}^t\mathbf{r}$ を推定する手法である。Kamvar らは、Aitken Extrapolation は式（4.21）と表せるという仮定の下での手法であることを再三強調している [32]。

Aitken Extrapolation は $\lambda_2$ と $\lambda_3$ とが互いに複素共役な固有値になっている場合にはあまり効果がない。そのため、Kamvar らは Aitken Extrapolation を拡張した Quadratic Extrapolation を同論文で提示している。Quadratic Extrapolation は、Aitken Extrapolation と同様に $k$ 回目の反復時の PageRank ベクトル ${}^t\mathbf{r}^{(k)}$ から $\lambda_2$ および $\lambda_3$ に対応する固有ベクトルを引くことで、真の PageRank ベクトル ${}^t\mathbf{r}$ を推定する手法である。

Kamvar らが行った 8,000 万の Web ページの PageRank 値をべき乗法で算出する実験では、通常のべき乗法よりも Quadratic Extrapolation を使ったべき乗法のほうが計算時間が短く、収束に至るまでに要した反復回数が少なくなったことが示されている（damping factor が 0.90 の場合には反復回数が約 35 回であったものが約 25 回に減っている）[32]。

### 4.4.4 BlockRank

Kamvar らは BlockRank[31] による PageRank 計算の高速化を提案している。以下で BlockRank を紹介する。

一般に、あるドメイン名の下にある Web ページの集まりは、それらのドメインに属する Web ページ同士で多くのリンクをし合う。例えば、www.ncsu.edu のドメインの下には多くの Web ページが存在し、それらの Web ページは www.ncsu.edu のドメインの下にある Web ページ同士で多くのリンクをし合っている。このように、互いに多くのリンクをし合っている Web ページの集まりをドメイン名などで区切ったものをホスト（Host）とよぶ。Web 上には、数千・数万以上のホストが存在するであろう。

BlockRank は、Web 全体をホストというブロックに分けて PageRank 値を算出することにより、一度に Web 全体の PageRank 値を求めるよりも効率的に計算を行う手法である。BlockRank の手順は以下の通りである。

(1) まず、ホスト内の Web ページの相対的な重要度を決定する PageRank 値を求める。これをローカル PageRank 値とよぶ。その際、ホスト外の Web ページとの

アウトリンクおよびインリンクについては無視するものとする。

(2) 次に、ホストを 1 つの Web ページと考え、全世界のホストから Web グラフを構成してそれにに対して PageRank 値を求める。その際、異なる 2 つのホストに属する Web ページ間のリンクだけを考慮するものとし、ホスト内の Web ページ同士でのリンクは無視する。ただし、ホスト間のリンクが 2 つ以上存在しても、1 つとして扱う。また同じホストに属する Web ページ間のリンクが存在しても無視するものとする（各ホストは Web ページと同様に自分自身へのリンクをもたないものとする。）。このようにして求めたホストに対する PageRank 値を HostRank 値とよぶ。HostRank 値はホストの数だけ算出される。

(3) 各ホストに属する Web ページのローカル PageRank 値にそれぞれの HostRank 値を乗じて、全世界の各 Web ページの PageRank 値とする。求めた Web ページ $k$ の PageRank 値を PageRank ベクトルの $k$ 番目の成分として記憶する。

Kamvar らの行った実験では、BlockRank による PageRank 値の計算に要した時間は、通常の PageRank 値計算の半分の時間であったことが示されている [31]。

## 4.5 PageRank 値の更新の高速化

本小節では、PageRank 値の更新 (再計算）の高速化についての研究を紹介する。4.4 節では PageRank 値の計算全般についての高速化の研究を紹介したのに対して、本小節では、過去の PageRank 値についての情報を活用することで、いかに効率的に新たな PageRank 値を算出するかをテーマにした研究を紹介する。

正確な PageRank 値を求めるにはクローリングを頻繁に行う必要があるという問題があるが、ここではハイパーリンク行列 **H** から PageRank 値を求める計算に問題を限定する。

まず「Web ページの更新」という言葉の意味を定義する。Web ページの更新を、リンク更新[*68]とページ更新[*69]との 2 つに分ける [38]。

**リンク更新：** Web ページ内のアウトリンクの追加・（または）削除が行われる。

**ページ更新：** Web ページ自体の追加・削除が行われるとともに、Web ページ内のアウトリンクの追加・削除が行われる。

リンク更新の場合、行列 **H** の行のうちアウトリンクの追加・削除が行われた Web ページの行の成分（リンク先 Web ページへの遷移確率）が変化する。しかし行列 **H** の大きさは変わらない。一方ページ更新の場合、行列 **H** の行自体の追加・削除が行われるため、ほとんどの場合、行列 **H** の大きさ自体が変化する。リンク更新は、ページ更新の 1 種と考えることもできる。以後、単に Web ページの更新といった場合にはリンク更新とページ更新とのいずれか、もしくは両方を指すものとする。

Web ページの更新と PageRank 値の更新との時間差は小さいほうがよいであろう。Cho らが 2000 年に行った実験 [12] では、データセット内の Web ページのうち約 40% の Web ページが一週間以内に更新されている。さまざまな Web ページが更新されつつある状況下では PageRank 値が更新されない期間が長いほど、重要な Web ページからのインリンクを集めた有用性の高い Web ページが新たに登場しても、その Web ページが検索結果に登場しない可能性が増える。

理想的にはどれか 1 つの Web ページが更新される毎に PageRank 値が更新されることが望ましいであろう。しかしクローリングを別としても PageRank 値の計算には多くの時間がかかるため、Google では PageRank 値の再計算は 1 ヶ月に一度の頻度で行ってい

[*68] link-updating。成分更新 (element-updating) ともよぶ [38]。

[*69] page-updating。状態更新 (state-updating problem) ともよぶ [38]。

る[*70]。Web ページの数は日々増え続けているため、少なくとも過去の PageRank 値を利用するなどして PageRank 値の更新を高速化しない限り、PageRank 値の更新間隔は大きくなってしまうであろう。

PageRank 値の更新はマルコフ行列 **G** の最大固有値 1 に対応する固有ベクトルの更新である。マルコフ連鎖の定常分布の更新は PageRank 値の更新以前から研究されてきたテーマであり、1980 年には Meyer らの研究 [44] がある。しかしながら [44] の手法はリンク更新にしか適用できず、かつ更新の前後で行列 **H** の成分が 1 行や 2 行程度しか変更されていない場合にしか適用できない手法であるため、PageRank 値の更新の場合にはあまり役に立たない [35]。

[*70] Google が PageRank 値の再計算を行う際に Web ページのランキングが動くことを指して、PageRank 値の更新は”Google Dance”ともよばれている [35]。

# 5 おわりに

本稿の目的は、検索エンジン Google の PageRank とそれに関連する研究を紹介するとともに、PageRank についての正しい理解を行うことであった。

本稿ではまず、2 節および 3 節で PageRank についての正しい理解を行うべく PageRank を紹介した。その内容を以下の 4 つにまとめる。

- PageRank は、Google が Web ページの評価に用いる要素の一つである PageRank 値を算出するアルゴリズムである。
- PageRank 値を算出することは、Web グラフ上を遷移するランダムサーファーの遷移確率を表すマルコフ行列の最大固有値 1 に属する左固有ベクトルを算出することに他ならない。
- 定常分布が常に一意的に算出できるように工夫されており、べき乗法はその主要な算出方法である。
- 1998 年の論文 [48] および [8] の PageRank 値の定義には食い違いや誤りがある。そのため、研究者によっては PageRank 値の再定義を行うことがある。

次に、4 節で PageRank に関連する研究の一部を紹介した。本稿では PageRank に関連する研究を 4 つのテーマに分類した。それぞれのテーマについて、以下のようにまとめる。

- PageRank 値の改良による検索結果の質の向上の手法を 4 種類紹介した。これらの手法では通常の PageRank に比べて PageRank 値の算出に使う情報が多くなるため、必要な記憶領域が大きくなる場合がある。また、クエリ従属なランキングアルゴリズムの場合にはクエリ応答時間が長くなる。PageRank 値の計算の高速化が実現すれば、算出できるパーソナライズド PageRank 値の数が増えて、検索結果の質の向上につながる可能性がある。
- PageRank 値の計算にかかる記憶領域の節約の手法として、隣接リストによるリンク情報の圧縮を紹介した。Web ページの総数が多くなればなるほど、その docID を効率的に記憶した場合に節約できる記憶領域は大きくなる。
- PageRank 値の計算の高速化の手法を 4 種類紹介した。それらの手法は Web グラフ上の状態を 1 つにまとめることで PageRank 値の計算問題のサイズを小さくする、べき乗法の反復回数を減らす、べき乗法の反復計算 1 回あたりの処理を軽減す

る、などの考え方に基づいている。

- PageRank 値の更新の高速化の手法として、マルコフ連鎖の定常分布の更新の手法を PageRank 値に応用する場合について述べた。しかし [44] の手法はリンク更新にしか適用できず、かつ更新の前後で行列 **H** の成分が 1 行や 2 行程度しか変更されていない場合にしか適用できないため、あまり実用的ではない。

筆者の反省点としては、最新の研究を網羅的に把握できていないことがある。本稿で扱った PageRank に関連する研究は、Web や論文書籍での閲覧が可能な研究の一部にすぎない。そのため、最新の研究ではより有用な手法が提案されている可能性がある。

本稿が PageRank およびそれに関連する研究について興味がある読者の助けとなれば幸いである。

# 謝辞

京都大学大学院人間・環境学研究科の櫻川貴司准教授に深く感謝の意を表します。櫻川貴司准教授には、不出来な筆者をときには厳しくときには優しく、また常に忍耐強く思慮深くご指導をいただきました。櫻川貴司准教授の多大なるご指導のおかげで本稿の執筆を終えることができました。

また、京都大学総合人間学部棟 1204 研究室の皆様および卒業生の皆様からは、本稿の執筆についての多数のご助言をいただきました。京都大学熊野寮の皆様からは、本稿を執筆する筆者への温かい励ましをいただきました。そして家族からは、本稿の執筆全般にわたり多大なるご支援をいただきました。ここに感謝の意を表します。

# 参考文献

[1] K. Avrachenkov and N. Litvak. The Effect of New Links on Google Pagerank. *Stochastic Models*, Vol. 22, No. 2, pp. 319–331, 2006.

[2] Ricardo Baeza-Yates and Emilio Davis. Web page ranking using link attributes. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pp. 328–329, New York, NY, USA, 2004. ACM.

[3] L.A. Barroso, J. Dean, and U. Hölzle. Web Search for a Planet: The Google Cluster Architecture. *IEEE MICRO*, pp. 22–28, 2003.

[4] Abraham Berman and Robert J. Plemmons. Nonnegative Matrices in the Mathematical Sciences (Classics in Applied Mathematics). Society for Industrial Mathematics, 1 1987.

[5] K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The Connectivity Server: fast access to linkage information on the Web. *Computer Networks and ISDN Systems*, Vol. 30, No. 1-7, pp. 469–477, 1998.

[6] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Trans. Interet Technol.*, Vol. 5, No. 1, pp. 92–128, 2005.

[7] S. Brin, R. Motwani, L. Page, and T. Winograd. What can you do with a Web in your Pocket? *Data Engineering Bulletin*, Vol. 21, pp. 37–47, 1998.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, Vol. 30, No. 1-7, pp. 107–117, 1998.

[9] Sergey Brin, Rajeev Motwani, and Terry Winograd. What can you do with a web in your pocket. *Data Engineering Bulletin*, Vol. 21, pp. 37–47, 1998.

[10] S. Chien, C. Dwork, R. Kumar, and D. Sivakumar. Towards exploiting link evolution. In *Workshop on Algorithms for the Web*, 2001.

[11] G.E. Cho and C.D. Meyer. Aggregation/Disaggregation Methods of Nearly Uncoupled Markov Chains.

[12] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 200–209, 2000.

[13] A. Clausen. Online Reputation Systems: The Cost of Attack of PageRank, 2003

[14] Jupitermedia Corporation. Teoma vs. Google, Round Two. http://dc.internet.com/news/print.php/1002061

[15] I. Drost and T. Scheffer. Thwarting the Nigritude Ultramarine: Learning to Identify Link Spam. *Lecture Notes in Computer Science*, Vol. 3720, p. 96, 2005.

[16] N. Eiron, K.S. McCurley, and J.A. Tomlin. Ranking the web frontier. In *Proceedings of the 13th international conference on World Wide Web*, pp. 309–318. ACM New York, NY, USA, 2004.

[17] R. Fagin, A.R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with" back buttons". *Annals of Applied Probability*, Vol. 11, No. 3, pp. 810–862, 2001.

[18] Gene H. Golub and Charles F. Van Loan. Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition). The Johns Hopkins University Press, 3rd edition, 10 1996.

[19] Google. Google の人気の秘密. http://www.google.co.jp/why_use.html

[20] Google. Google サポート ウェブ履歴について. http://www.google.com/support/accounts/bin/answer.py?answer=54068&topic=14149

[21] Google. リンク プログラム - ウェブマスター向けヘルプ センター. http://www.google.com/support/webmasters/bin/answer.py?answer=66356

[22] Z. Gyongyi and H. Garcia-Molina. Web Spam Taxonomy. *Adversarial Information Retrieval on the Web*, 2005.

[23] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pp. 517–528. VLDB Endowment, 2005.

[24] T.H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the Eleventh International World Wide Web Conference*, pp. 517–526, 2002.

[25] T.H. Haveliwala. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge nad Data Engineering*, pp. 784–796, 2003.

[26] T.H. Haveliwala, et al. Efficient computation of PageRank. *Stanford University,*, Vol. 8090, pp. 1998–31, 1999.

[27] T.H. Haveliwala and S.D. Kamvar. The second eigenvalue of the Google matrix. *A Stanford University Technical Report*, Vol. 8090, pp. 2003–20.

[28] M. Haviv. Aggregation/Disaggregation Methods for Computing the Stationary Distribution of a Markov Chain. *SIAM Journal on Numerical Analysis*, Vol. 24, p. 952, 1987.

[29] J. Hirai, S. Raghavan, H. Garcia-Molina, and A. Paepcke. WebBase: a repository of Web pages. *Computer Networks*, Vol. 33, No. 1-6, pp. 277–293, 2000.

[30] American Customer Satisfaction Index. ACSI Quarterly Scores Q2 2008. http://www.theacsi.org/index.php?option=com _content&task=view&id=184

[31] S.D. Kamvar, T.H. Haveliwala, C.D. Manning, and G.H. Golub. Exploiting the block structure of the web for computing pagerank. *Preprint, March*, 2003.

[32] S.D. Kamvar, T.H. Haveliwala, C.D. Manning, and G.H. Golub. Extrapolation methods for accelerating PageRank computations. In *Proceedings of the 12th international conference on World Wide Web*, pp. 261–270. ACM New York, NY, USA, 2003.

[33] J.M. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A.S. Tomkins. The Web as a Graph: Measurements, Models and Methods. *Lecture Notes in Computer Science*, pp. 1–17, 1999.

[34] J.O.N.M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, Vol. 46, No. 5, pp. 604–632, 1999.

[35] Amy N. Langville and Carl D. Meyer. Google's Pagerank and Beyond: The Science of Search Engine Rankings. Princeton Univ Pr, 7 2006.

[36] A.N. Langville and C.D. Meyer. Deeper Inside PageRank. *Internet Mathematics*, Vol. 1, No. 3, pp. 335–380, 2004.

[37] A.N. Langville and C.D. Meyer. A Reordering for the PageRank Problem. *Siam Journal on Scientific Computing*, Vol. 27, No. 6, p. 2112, 2006.

[38] A.N. Langville and C.D. Meyer. Updating Markov Chains with an Eye on Google's PageRank. *Siam Journal on Matrix Analysis and Applications*, Vol. 27, No. 4, p. 968, 2006.

[39] C.P.C. Lee, G.H. Golub, and S.A. Zenios. A fast two-stage algorithm for computing PageRank and its extensions. *Scientific Computation and Computational Mathematics*, 2003.

[40] F. Mathieu and M. Bouklit. The effect of the back button in a random walk: application for pagerank. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pp. 370–371. ACM New

York, NY, USA, 2004.

[41] George Meghabghab and Abraham Kandel. Search Engines, Link Analysis, and User's Web Behavior: A Unifying Web Mining Approach (Studies in Computational Intelligence) (Studies in Computational Intelligence). Springer, 1 edition, 4 2008.

[42] Carl Meyer. Matrix Analysis and Applied Linear Algebra Book and Solutions Manual. SIAM: Society for Industrial nad Applied Mathematics, 2 2001.

[43] C.D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Rev*, Vol. 31, No. 2, pp. 240–272, 1989.

[44] C.D. Meyer and J.M. Shoaf. Updating finite markov chains by using techniques of group matrix inversion. *Journal of Statistical Computation and Simulation*, Vol. 11, No. 3, pp. 163–181, 1980.

[45] Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. *ACM Comput. Surv.*, Vol. 28, No. 1, pp. 33–37, 1996.

[46] I.S. Nathenson. Internet Infoglut and Invisible Ink: Spamdexing Search Engines with Meta Tags. *Harvard Journal of Law & Technology*, Vol. 12, p. 43, 1998.

[47] L. Page. Method for node ranking in a linked database, June 6 2006. US Patent 7,058,628

[48] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998

[49] S. Raghavan and H. Garcia-Molina. Representing Web graphs. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pp. 405–416, 2003.

[50] M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. *Advances in Neural Information Processing Systems*, Vol. 2, pp. 1441–1448, 2002.

[51] T. Suel and J. Yuan. Compressing the graph structure of the web. In *Data Compression Conference (DCC)*, pp. 213–222, 2001.

[52] M. Sydow. Random surfer with back step. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pp. 352–353. ACM New York, NY, USA, 2004.

[53] Mike Theiwall. Link Analysis: An Information Science Approach (Library and Information Science). Academic Press, 12 2004.

[54] K. Thorson. Modeling the Web and the Computation of PageRank. *Undergrad-*

*uate thesis, Hollins University*, 2004.

[55] B. Wu and B.D. Davison. Identifying link farm spam pages. In *International World Wide Web Conference*, pp. 820–829. ACM New York, NY, USA, 2005.

[56] Baoning Wu and Brian D. Davison. Identifying link farm spam pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pp. 820–829, New York, NY, USA, 2005. ACM.

[57] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. Van Roy. Making Eigenvector-Based Reputation Systems Robust to Collusion. *Lecture Notes in Computer Science*, pp. 92–104, 2004.

[58] F. シャトラン. 行列の固有値 - 最新の解法と応用. シュプリンガーフェアラーク東京, 新装版, 6 2003.

[59] R. セジウィック. アルゴリズム C〈第 1 巻〉基礎・整列. 近代科学社, 9 1996.

[60] 森村英典, 高橋幸雄. マルコフ解析 (OR ライブラリー 18). 日科技連出版社, 3 1979.

[61] 西田圭介. Google を支える技術 巨大システムの内側の世界 (WEB+DB PRESS プラスシリーズ). 技術評論社, 3 2008.

[62] 豊田秀樹. マルコフ連鎖モンテカルロ法 (統計ライブラリー). 朝倉書店, 5 2008.

[63] 福島正俊, 竹田雅好. マルコフ過程 (確率論教程シリーズ). 培風館, 2 2008.

[64] 伊理正夫. 一般線形代数. 岩波書店, 2 2003.

[65] 総務省郵政事業庁. 平成 9 年 通信に関する現状報告. http://www.johotsusintokei.soumu.go.jp/joho_tsusin/policyreports/japanese/papers/97wp1.html

[66] みずほ情報総研株式会社 吉川日出行. サーチアーキテクチャ 「さがす」の情報科学. ソフトバンククリエイティブ, 9 2007.

[67] 神崎洋治, 西井美鷹. 体系的に学ぶ検索エンジンのしくみ. 日経 BP ソフトプレス, 12 2004.