WWW のリンク構造の解析

Analysis of the link structure of WWW

http://www.tani.cs.chs.nihon-u.ac.jp/g-2003/nobu
http://www.tani.cs.chs.nihon-u.ac.jp/g-2003/hiro

谷　研究室　　清水　伸明・末次　寛之
Nobuaki SHIMIZU, Hiroyuki SUETSUGU

概要

WWW のリンク構造をグラフとみなすことにより，リンク構造の解析に離散アルゴリズム的な手法が用いられるようになった．例えば HITS，PageRank などのアルゴリズムはその一例である．本研究では, これらの手法を用いて，権威のあるページを探すシステムを試作した.

1　はじめに

近年インターネットの普及により,WWW(World Wide Web) は急激に成長をつづけてきた. しかしこのネットワークインフラは鉄道や道路などのインフラと違い, 常に変化しているためどのような広がりを見せるか予測し難い. その結果,WWW には大量の情報が氾濫しており, そこから効率よくよい情報を得ることが難しくなってきている．そこで, 情報を扱いやすくするための方法が多く試みられている．１つの方法として, 探したいトピックに対してコミュニティーを作り, そのリンク構造からよい情報を探しだすものがある. その中で有名なものに,HITS や PageRank がある．本研究ではこの２つのアルゴリズムを取り上げ権威のあるページを探しだすシステムを試作する.

本稿では, 以下の内容で構成される. まず２章では HITS という手法について述べ，３章では Google で採用されている PageRank という手法を述べる. そして４章以降では，実装，実験の方法や実験結果の例などを示し，今後の展望について述べる事とする.

2　HITS

この章では, まず Hub,Authority に関する用語を解説し, Jon M.Kleinberg が考案した HITS を紹介する.

2.1　HITS の概要

トピックに深く関連した内容を含む Authority ページと, 多くの Authority ページへのリンクを持つ Hub ページというものを定義し, それらの相互関係を利用して各ページ Authority 値と Hub 値を求めることで重要ページを探し出すものである．Authority とは, それ自身が情報を提供しているページで多くのサイトからリンクを張られている.Hub とは多くの Authority をリンクする. リンク集的なページである.(図１) この手法が対象とする Web グラフは，あるトピックに関連したページ群であり, そのページ間のリンクのほとんどは意味的なリンクであると考えられる. つまり各ページからのリンクは, そのページとの関連があって重要度の高いページへのリンクが多いと考えられる.

図 1：Hubs & Authority

2.2　HITS アルゴリズム

STEP1 抽出ステップ（A sampling step）

探したいクエリに対し関係すると思われる Web ページを約２００ページ探し, その２００のページからリンクを何度かたどり１０００～３０００のページを集める.

STEP2 重さ伝播ステップ（A weight-propagation step）

抽出ステップで集めたサイトを隣接行列にし Hub や Authority に重みをつけていく．そして重みの重い Hub や Authority を見つける.

2.2.1　重みのつけ方

もし１つのページが沢山のよい Hubs にリンクされているならば, Authority weight を増やす. ページ p にリンクしているすべてのページ q の集合を y_q として, y_q

の合計が x_p の値になる.

$$x_p = \sum_{q \ s.t. \ q \to p} y_q$$

同じように, 沢山のよい Authority にリンクを貼っている Hub の weight を増やす. ページ p にリンクしているすべてのページ q の集合を y_q として,y_q の合計が x_p の値になる.

$$y_p = \sum_{q \ s.t. \ q \to p} x_q$$

これらの式をより数学的に更新できる式を以下に示す. ページ {1,2,...,n} があるとする. それらのリンク構造から対応する n × n の行列 A を次のように定義する: あるページ u からページ v へリンクが張られている場合には $A_{u,v}$=1 とし, リンクがない場合には $A_{u,v}$=0 とすることにより定まる隣接行列. $x = (x_1, x_2, ..., x_n)$, $y = (y_1, y_2, ..., y_n)$, を n 次ベクトルとする. 更新ルールを $x \leftarrow A^T y$, $y \leftarrow Ax$ として Authority weight と Hub weight をもとめる式を次のように定義する:

$$x \leftarrow A^T y \leftarrow A^T Ax = (A^T A)x$$

$$y \leftarrow Ax \leftarrow AA^T y = (AA^T)y$$

3 Google - PageRank -

この章では, ラリー・ページ（Larry Page）とサージ・ブリン（Sergey Brin）がスタンフォード大学在学中に提案した PageRank はリンク構造に基づいて Web ページのランキングを計算するアルゴリズムである.

3.1 PageRank の概要

あるページ A からあるページ B へのリンクをページ A からページ B への支持投票とみなし（図 2）, その票数からそのページの重要度を算出する. この考え方自体は, 論文の引用率に基づいた評価や計量社会学において１９５０年代から存在する. PageRank 法を用いれば, 検索結果の上位に有名な Web ページ, すなわち他のサイトから沢山のリンクが張られているページを表示できる. 例えば,「文学部」で検索すると, 第 1 位に早稲田大学, 第 2 位に慶応義塾大学, 第 3 位に京都大学が検索結果として表示されるが, これは, 早稲田大学文学部のホームページが他の沢山のサイトからリンクが張られ有名であることを示す.

図 2：PageRank のイメージ

3.1.1 PageRank の利点

これまでにも張られているリンク (Back Link) 数を重要度として用いるサーチエンジンは存在したが, PageRank 法では, リンク元であるページＡの重要度によって, １票の重みを変えるためそれほど重要ではないページＣが無意味なページから多数のリンクを受けているような場合には, ページＣの重要度が増してしまうのを避けることができる. サーチエンジンを使うユーザからみると, 他の多くの人が良いと思ってリンクを張るような Web ページを効率よく探すことができ, 検索に要する時間の短縮が可能となる.

3.2 PageRank アルゴリズム

3.2.1 計算式

基本的な PageRank の基本概念は前述した通りだが, この節ではその内容を具体的な計算式として示すことにする. $Rank(v)$ を v の $PageRank$, $Rank(u)$ をページ v へのリンクをもつのページ u の $PageRank$ とする. N を対象とするグラフ G のノードの総数（Web ページ数）, Nu をページ u からの外向きのリンク数として Bv は v を指しているページの集合とする. c を dampening factor と呼び,$0 < c < 1$ の定数だが通常 0.1〜0.2 に設定される. PageRank は以下の式を繰り返し適用することにより計算される.

$$Rank(v) = c\frac{1}{N} + (1-c)\sum_{u \in B_v} \frac{Rank(u)}{N_u}$$

3.2.2 行列計算による方法

前節で述べた計算式を実際に全てのページに適用するために, 行列を用いて計算を行う. 計算は以下の方法によって計算される.

まず, 前の章同様に u を Web ページ,u からリンクされているページの集合を F_u,N_u を u から出ているリン

クの数, c を一般化のための定数とする. この時, ある行列 A を考える. この行列は要素 $A_{u,v}$ を次のように定義した行列である. もし u から v へとリンクが貼られている場合には $A_{u,v}$=1/Nu, もしリンクが無い場合には $A_{u,v}$=0 となる. そして,R をそれぞれの Web ページを持つ $PageRank$ の値の要素とするベクトルとした場合に

$$R = cAR$$

という式を得る. このRは行列Aの固有ベクトルとなることがわかっている. 実際にはRに対してAを繰り返し適用することによって値を計算することになる.

4 Web 情報統合技術の実装

一般に Web 情報統合は，以下のようなプロセスから成り立っている.

1. 必要な情報が記述されている Web ページを収集する.
2. 収集した Web ページから必要な情報を抽出する.
3. 抽出した情報を関連付ける.

情報統合の過程を手続き的に記述する技術として ，WebSQL(http://www.cs.toronto.edu/~websql), WebL(http://www.research.compaq.com/SRC/WebL)) などが開発されているが，ここでは，情報源の選択やアクセスの順序付けを行うプランニング (Planning) をより強化することを目的とし，独自のツール (以下 webagent) を作成することとした.

我々が開発した webagent では Web をドキュメント（ページ）に関するデータベースとしてモデル化しており，各ドキュメント (Document) には，URL，Address(IP Address)，タイトル (Title)，文章 (Content)，タイプ (ContentType)，長さ (ContentLength)，ドキュメントに含まれるリンク数 (LinkCount) の属性が記述されている.

情報の収集は，エージェントにより実行される．適切なインターフェースに SQL ライクなクエリを送信すると，検索方法，検索条件が親となるエージェントに伝えられる．それに基づき親エージェントが Todo リストを作成し，子となるエージェントを生成し，子元にエージェントがを Todo リストに従い情報の収集を行う.

親エージェントに集められたドキュメントは必要に応じ解析され，再帰的に Todo リストに次の指令を追加する.

webagent では次のようなクエリが有効である.

```
SELECT TOP 1000 *, -Content
 FROM http://www.chs.nihon-u.ac.jp/
 WHERE
  BreadthFirst
   URL CONTAINS “chs.nihon-u.ac.jp” AND
   ContentType CONTAINS “text”>>{0,3}
```

SQL はデータベースを操作する言語として，最も一般的に利用されているものであり，データ検索を行う最も基本的な SQL 文は SELECT 句，FROM 句，WHERE 句から構成される.

webagent では，SELECT 句には表示すべき属性，FROM 句には検索のスタート地点となる URL，WHERE 句には検索の条件を指定する.

ここで，WHERE 句の最後にある >>{3}は,「ドキュメントに含まれる任意のリンクを 3 階層まで検索する」ことをあらわしている.

このように，Web を巡回するためのオペレータとしては “->” （同一ホスト），“=>”（異なるホスト）があり，それぞれに対し，->{m,n} または->+ ({1,} と同意味, n 省略時は無限回)， ->* (={0,})，->?(={0,1}) のように検索対象となるドキュメントの階層を指定することができる.

AND，OR，XOR，NOT を含む一つのステートメント (Statement) は->，=>，>>により連続して指定することができ，各ステートメントの最初には，BreadthFirst（幅優先探索），DepthFirst（深さ優先探索）のキーワードを用いて，検索方法を選択する.

演算子は CONTAINS(含む) のほかに，MATCHES(正規表現によるマッチング)，EQUALS(等しい)，<，>，<=，>=，!=，==（数値による比較）を用いることができる.

また，Google Web APIs（http://www.google.com/apis/）を用いることで，FROM 句に既存のサーチエンジンを利用し，以下のようなクエリも可能である.

```
SELECT URL, Title
 FROM google?”keywords”
 WHERE BreadthFirst ->*
```

データ収集が目的の場合，INTO 句を用いて以下のようなクエリでローカルディレクトリにドキュメントを保存することもできる.

```
SELECT URL
 INTO /home/foo/path/
 FROM http://www.yahoo.co.jp/
```

5 実験と考察

webagent の収集したデータを元に，Jama.1.0.1 (http://math.nist.gov/javanumerics/jama/) を利用し，Authority Weight， Hub Weight，PageRank をそれぞれ算出した．

検索対象とした多くのネットワークでは，共通して，高い Hub Weight を持つ，極めて少数のノードの存在が確認された．

ランダムな隣接行列を用いての実験では，大多数のノードが同様の Hub Weight を持ち，所持するリンク数とノードの数は，ほぼ正規分布に従う結果がでており，それに対し，Web 上での Hub は指数分布的に存在している．

このような Hub は，所属するネットワークにおいて，重要度の高い Authority へのリンクを持つという性質の他に，その自体の存在が，他の多数へのノード同士を結びつける重要な役割を果たしていることが容易に想像でき，また，その Hub を取り除くことによって，多くの優良な情報を持つ Authority へたどり着けない，あるいは，到達するのにより多くの経路が必要になることが考えられる．

先に述べた webagent 利用し，日本大学文理学部 (http://www.chs.nihon-u.ac.jp) をルートとした学部内のリンク構造を解析した結果の例を別途付録に示す．

参考文献

[1]Jon M. Kleinberg[1] ,Ravi Kumar[2] ,Prabhakar Raghvan ,Sridhar Rajagopalan[2] ,and Andrew S. Tomkins[2], *On The Web as a graph: measurements, models, and methods*, [1] Department of Computer Science, Cornell University, Ithaca, NY 14853. [2] IBM Almaden Research Center K53/B1, 650 Harry Road, San Jose CA 95120.

[2]Taher H. Haveliwala, *Efficient Computation of PageRank*, Stanford University, taherh@db.stanford.edu

[3]L. Page and and , $S.Brin$,R. Motwani, $andT.Winograd$, *The PageRnk Citation Ranking: Bringing Order to the Web*, 1998

[4]Albert-Laszlo Barabasl, *LINKED*: *The New Science of Networks*, 2002

付録 実験結果

順位	Authority Weight	URL	Title
1	0.267	http://www.chs.nihon-u.ac.jp/	文理学部へようこそ！
2	0.262	http://www.chs.nihon-u.ac.jp/entrance.html	日大文理 入学案内
3	0.151	http://www.chs.nihon-u.ac.jp/syllabus/	平成 14 年度授業計画

順位	Hub Weight	URL	Title
1	7.243	http://www.chs.nihon-u.ac.jp/ninfo.html	総合案内
2	2.534	http://www.chs.nihon-u.ac.jp/dept.html	学科紹介
3	2.371	http://www.chs.nihon-u.ac.jp/index-con/navi_dept.html	日大文理学部

順位	PageRank	URL	Title
1	8.213	http://www.chs.nihon-u.ac.jp/panph/index.html	日大文理学部
2	8.154	http://www.chs.nihon-u.ac.jp/index-con/entrance$_f$.*html*	日大文理学部
3	8.121	http://www.chs.nihon-u.ac.jp/voice/index.html,	日大文理学部