類似値を用いたWWWのリンク構造の解析

Analysis of the link structure of WWW using the similar value

http://www.tani.cs.chs.nihon-u.ac.jp/g-2004/kuri/

谷 研究室　栗原　伸行

Nobuyuki KURIHARA

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 や PageRank 等のアルゴリズムはその一例である。本研究では、HITS アルゴリズムを改善手法を提案し、その有効性を検証するための比較実験を行う。

1　はじめに

近年インターネットの普及により WWW(World Wide Web) は拡大の一途をたどり続けており、2003 年 1 月現在、全世界の総ドメイン数は 171,638,297 と報告させている。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は違い、常に変化しているためどのように広がりを見せているか予想し難い。その結果、そこから効率よくより良い情報を得ることが難しくなっている。そこで、情報を扱いやすくする為の方法が多く試みられている。Web 上で独立したページが何らかの意図を持って互いのページにリンクを張り合うという特徴のリンク構造”Web コミュニティ”に注目し、より良い情報を探し出すものもその一つであり、その中で有名なものとして、HITS や PageRank などがある。本研究では前者の HITS アルゴリズムを取り上げ改善手法を提案しその有効性を評価するため比較実験を行う。今回提案する手法として、ページ間に類似値を用いるものとトピック値を用いるもの、この 2 つを述べる。

本稿では、以下の内容で構成される。まず 2 章では HITS アルゴリズムの手法について述べ問題点を挙げる。そして 3 章以降では、改善を施したアルゴリズムの手法を述べ、今後の展望を述べることとする。

2　HITS(Hyperlink-Induced Topic Search)

この章では、Web 上のリンク構造を解析し、Authority と Hub と呼ばれる 2 つのページ集合から信頼性の高い情報を抽出した、Jon M.Kleinberg の HITS(Hyperlink-Induced Topic Search) アルゴリズムを紹介する。

2.1　HITS アルゴリズムの概要

HITS アルゴリズムの特徴は、各 Web ページのコンテンツ内容には立ち入らずページ間のリンク情報の解析だけで、あるクエリに対して適切な情報 (Authority と Hub の 2 つのページ集合) を得ることが出来ることにある。Authority とは、あるトピックにおいて的確な情報を提供しているページであり、多くのページからリンクを張られている。また 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。この手法が対象とする Web コミュニティは、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。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ている。

2.2　HITS アルゴリズムの手順

STEP1

探したいクエリに対し関係すると思われる Web ページを一定数 r 件収集し root set とする。

STEP2

root set に含まれるページからリンクされている全てのページ、及び root set に含まれるページにリンクしているページ最大 d 件を収集し root set に追加、大きさ n の base set を作成する。

STEP3

base set のページについて Authority や Hub に重みをつけていく。

なお、Kleinberg は r を 200、d を 50 に設定し、その結果 n が約 1000〜5000 ぐらいになると報告している。

2.2.1　重みの付け方

もし 1 つのページが沢山のよい Hub にリンクされているのならば、Authority weight を増やす。ページ p にリンクしている全てのページ q の集合を y_q として、y_q の合計が x_p の値となる。

$$x_p = \sum_{qs.tq \to p} y_q$$

同じように沢山の良い Authority にリンクを張っている Hub の weight を増やす。ページ p にリンクされている全てのページ q の集合を x_q として、x_q の合計が y_p の値となる。

$$y_q = \sum_{qs.tq \to p} x_p$$

これら 2 式を反復して weight を抽出する。両者の関係は、よい authorty は複数の良質の Hub によってリンクされ、また良質の Hub は複数のよい Authority にリンクを張っている。と、再帰的に定義している。これら 2 式より数学的に更新できる式を示すが、結局これらはページ間の隣接行列に関連する固有値問題である。

隣接行列

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

$$a_{ij} = \begin{cases} 1 & \text{if page i point to page j} \\ 0 & \text{otherwise} \end{cases}$$

反復計算

$X_i = (x_1, x_2, \cdots, x_n)$ $Y_i = (y_1, y_2, \cdots, y_n)$ を n 次ベクトルとし、$X_0 = (1, 1, \cdots, 1)$ $Y_0 = (1, 1, \cdots, 1)$ とする。そして、更新ルールを、$X_{i+1} \leftarrow A^T Y_i$ and $Y_{i+1} \leftarrow A X_{i+1}$ とし収束するまで行う。

$$X_k = (A^T A)^{k-1} A^T X_0 \quad (1)$$

$$Y_k = (A A^T)^k Y_0 \quad (2)$$

上記より (1)(2) の 2 式が求められるが、$A^T A$ は $n \times n$ の対象行列であるため、$X_\infty = A^T A X_0$の最大固有値に対応する固有値ベクトル 、$Y_\infty = A^T A Y_0$の最大固有値に対応する固有値ベクトルである。

2.2.2 問題点

HITS アルゴリズムは、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。という前提のもとで行われている。しかし、現在の Web コミュニティは必ずしもそうとは限らない場合がある。つまり、内容のまったく異なるページにリンクをしていることも多々あるということである。そのため、base set に本来のトピックにはまったく関係のないページが含まれ、そうしたページが密なリンク構造である場合、探したいトピックとはかけ離れたページに高い Weight を与えてしまう。これが topic drift 問題である。

3 HITS アルゴリズムの改善

この章では前章で挙げられた問題に対し修正法を用いてアルゴリズムを改善する。

3.1 HITS アルゴリズムの改善点の概要

HITS アルゴリズムは、Web ページの内容に立ち入らないことが前提となっており、あるトピックに関連のないページに対しても関連があるページと同様にあつかってしまう。そこで、本研究では以下のことを提案する。

(i) ページの内容に立ち入り、リンクを貼ページを貼られるページの類似度をとり、その値を隣接行列に使用する。

(ii) リンクを貼られているページ内で探しているトピックが含まれているならば、一定の値を与える。

リンクを貼るページと貼られるページの類似値が極端に低いのであれば探しているトピックに関連しているとは考えにくく、探しているトピックが本文中に含まれているならば、そのページの weight をおもくして良いと考えられるからである。

3.2 各ページの類似値

2 つのページの類似値を取るために、次の方法において行う。索引語を抽出し、その各索引語において重み付けを行う。そして各ページをベクトル空間モデルで表記し、2 つベクトル空間においてコサイン尺度をとることで類似値を求める。

3.2.1 索引語の抽出

Web ページのコンテンツ内容を、文書中に含まれる単語の集合で近似する事とする。しかし、文書に含まれる単語すべてが一律に文書の内容と関係しているわけではない。たとえば、助詞、助動詞などに当てはまる。したがって文書の内容を特徴づけるための単語を抽出する必要があり、その単語を索引語と呼ぶ。本研究では、各 Web ページの文書に形態素解析を行い、文書を単語に分割し、名詞、動詞、英単語を抽出する。なお、形態素解析には茶筌を使用し、動詞は基本形に直し使用する。

3.2.2 索引語の重み付け

前目では、索引語を抽出する処理について述べた。しかし、索引語の中には、文書の内容と密接に関係したものもあれば、関係の薄いものも存在するので、その索引語がどれだけ重要度を持っているかを図るために重み付けを行う。いま、n 個のページ $D_1, D_2..., D_n$ があり、これらの文書集合から全部で m 個の索引語 $w_1, w_2, ..., w_m$ が抽出されたとする。このとき索引語 w_iのページ D_j における重み d_{ij} は、以下のように示す。

$$d_{ij} = TF_{ij} \cdot IDF_i$$

TF**(索引語頻度 trem frequency)**

索引語 w_i のページ D_j における出現頻度に基づき計算される重みである。文書中に頻繁に出現する索引語に大して大きな値を与える。

$$TF_{ij} = \frac{(\text{ページ } D_j \text{において索引語 } w_i \text{の出現個数})}{(\text{ページ } D_j \text{において出現する全単語数})}$$

IDF**(文書頻度の逆数 inverse document frequency)**

文書集合全体にわたる索引語 wi の出現頻度の偏りを考慮して決定される重みである。特定の文書に集中して出現する索引語に対して大きな値を与える。

$$IDF_i = \log \frac{(\text{文書集合中の文書の総数})}{(\text{索引語 } wi \text{ を含む文書数})}$$

3.2.3 ベクトル空間モデルを用いた類似値

今目では、前目までに求めた索引語の重みを要素とするベクトルで文書を表現する。いま対象となる Web ページを $D_1, D_2, ..., D_n$ として、これら全体を通じて全部でm個の索引語 $w_1, w_2, ..., w_m$ があるとする。このとき D_j は次のようなベクトルで表現することができる。

$$d_j = \begin{pmatrix} d_{1j} \\ d_{2j} \\ \vdots \\ d_{mj} \end{pmatrix}$$

ここでの dij は索引語 wi のページ Dj における重みである。ここから、各ベクトルにおいて類似度を求めるが、本研究ではコサイン尺度を用いる。

$$\cos(d_j, d_k) = \frac{d_j \cdot d_k}{\|d_j\| \|d_k\|}$$

$$= \frac{\sum_{i=1}^{m} d_{ij} d_{ik}}{\sqrt{\sum_{i=1}^{m} d_{ij}^2} \cdot \sqrt{\sum_{i=1}^{m} d_{ik}^2}}$$

これにより、各ページ同士の類似値をもとめることができる。

3.2.4 隣接行列の改善

2．2．1 で示した隣接行列を以下のように改善を施す。

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

$$a_{ij} = \begin{cases} cos(d_i, d_j) & \text{if page i point to page j} \\ 0 & \text{otherwise} \end{cases}$$

以降は、2．2．1 と同様な手順で進む。

3.3 トピック値

リンクが貼られているページにおいて、自分が探しているトピックがページ中にあれば一定の値を与える。

3.3.1 トピック値の決定

本来ならば数多くのトピック値で実験を行い、統計をとり、最適なトピック値を決定しなければならないが、今回十分な実験を行えなかったため、一番良かったと思われる値を使用する。

$$topic(d_j) = \begin{cases} 1 + cos(d_i, d_j) & \text{if topic } \in d_j \\ 1 & \text{otherwise} \end{cases}$$

3.3.2 隣接行列の改善

2．2．1 で示した隣接行列を以下のように改善を施す。

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

$$a_{ij} = \begin{cases} cos(d_i, d_j) \cdot topic(d_j) & \text{if page i point to page j} \\ 0 & \text{otherwise} \end{cases}$$

以降は、2．2．1 と同様な手順で進む。

4 比較実験と考察

4.1 実験方法

今回の実験には以下の 3 つの方法を用いて比較実験を行った。

(i) 従来の手順で行う HITS

(ii) 隣接行列に類似値を用いたもの

(iii) 隣接行列に類似値・トピック値を用いたもの

そして、各パラメータ r=100 (root set のサイズ)、n = 50(1 ページあたりの最大リンク数) とした。root set の収集方法として、URL を 1 つ入力しその URL からリンクをたどり集める。たどる順番は幅優先とした。専門性の高いページを指定することで、そのページから同種のページにリンクが貼られていると仮定しているため、この収集方法を用る。

4.2 実験結果

別途表参照

4.3 考察

今回先に述べた実験を行い、いくつかのケースをのぞき、目に見えた結果、差を得ることができなかった。このような実験結果を得て、以下のようにまとめる。

- WWW における情報検索には、ページ内容に立ち入らずリンク構造の解析だけで十分判断可能である。

ということが分かる。しかし、数は少ないがいくつかの例外的結果も出ていることから、

- 類似値・トピック値それぞれにおいて十分なチューニングを行うことで精度の向上の可能性がある。

ということも、提案してまとめとする。

4.4 今後の改善

考察においてチューニングの必要性をあげたが、ここでは具体的なチューニング方法を述べる。

4.4.1 低次元空間への射影

類似値を求めるにあたり Web ページをベクトルに表したが、ベクトル空間モデルでは文章ベクトルの次元数は索引語の総数と等しい。したがって、検索対象をなるページが増えるに従い、ベクトルの次元数も増加する傾向が見られた。次元数が増加してくると、文章中に含まれる不必要な索引語がノイズ的な影響を及ぼし、検索精度を低下させてしまう可能性がある。そこで、高次元空間にある文章ベクトルを低次元の空間に射影してみることを提案する。これにより、今までは別々に扱われていた索引語が、低次元の空間では相互に関連を持ったものとして扱われる可能性があり、索引語の持つ意味や概念に基づく検索を行うことができるかもしれない。

4.4.2 文章検索の手法の有効性

今回、類似値を求めるにあたり文章検索でよく用いられるポピュラーな手法を使用したが、本新聞などの文章と Web ページの文章とは質が違うため手法の有効性に疑問が残る。例えば文章における偏りを調べる際に、文章の区切りを句読点ピリオドで判別したが、Web ページを見る限りでは曖昧なことが多い。また、Web ページの 1 文は本新聞に比べて短いことも違いとして上げられる。これより、索引語の重み付けを行う際、TF・IDF などの手法をそのまま用いるのではなく、Web ページ用にチューニングをすべきである。

4.4.3 今回の実験における root set・base set の収集方法

今回、root set の収集方法は、幅優先を使用し 1 ページのリンク数５０超えた時その後のリンクにおいては切り捨てたが、収集の方法を変化することで (ランダムなど) 違う root set を収集してみるべきである。また、今実験において base set の収集方法は root set から深さ 1 の範囲を対象にしていたが、深さを広げることでまた違った結果を得ることができるのではないかと考える。

参考文献

[1] Jon M.Kleinberg, Authoritative sources in a hyperlinked environment, IBM, 1998

[2] 北 研二, 津田 和彦, 獅々堀 正幹, 情報検索アルゴリズム, 共立出版, 2002

[3] 茶筌, 奈良先端科学技術大学院大学情報科学研究科自然言語処理学講座 (松本研究室), http://chasen.aist-nara.ac.jp/

[4]Jama,A Java Matrix Pakeage, http://math.nist.gov/javanumberics/java/

[5]Internet Domain Survey, http://www.isc.org/ds/

実験結果

1) URL : http://math.nist.gov/javanumerice/jama/

・HITS

順位	URL
1	$http://www.netlib.org/lapack/index.html$
2	$http://www.netlib.org/linpack/readme$
3	$http://www.mathworks.com$

・類似度を用いたもの

順位	URL
1	$http://www.mathworks.com/company/pressroom/index.shtml/article/439/index.shtml$
2	$http://www.mathworks.com/company/pressroom/index.shtml/article/439/siteindex.shtml$
3	$http://www.mathworks.com/company/pressroom/index.shtml/article/439/search$

・トピック値を用いたもの (対象とするトピック : Matrix)

順位	URL
1	$http://www.mathworks.com/company/pressroom/index.shtml/article/435/index.shtml$
2	$http://www.mathworks.com/company/pressroom/index.shtml/article/435/siteindex.shtml$
3	$http://www.mathworks.com/company/pressroom/index.shtml/article/435/search$

2) URL : http://www.pure.cc/~winds/volleyball/sunflower/main.html

・HITS

順位	URL
1	$http://www.jva.or.jp/jva/schedule.html$
2	$http://www.jva.or.jp/topics/index.html$
3	$http://www.jva.or.jp/jva/index.html$

・類似度を用いたもの

順位	URL
1	$http://www.jva.or.jp/japan/motoko/20040127.html$
2	$http://www.jva.or.jp/japan/motoko/20031224.html$
3	$http://www.jva.or.jp/japan/motoko/20031210.html$

・トピック値を用いたもの (対象とするトピック : 栗原 恵)

順位	URL
1	$http://momocan1111.hp.infoseek.co.jp/megu/index.html$
2	$http://momocan1111.hp.infoseek.co.jp/azusa/redrockets\_members.html$
3	$http://momocan1111.hp.infoseek.co.jp/megu/vleague.html$