

Kolmogorov 記述量に基づいた類似度を用いた WWW リンク構造解析手法の提案

A proposition of an algorithm for analyzing WWW-link structure with
Kolmogorov complexity

<http://www.tani.cs.chs.nihon-u.ac.jp/~joykun/>

谷 研究室 真板 育海
Ikumi Maita

概要

近年のネットワーク技術の発展により、ネットワーク上には多くの情報が溢れている。そのような状況で効率よく有益な情報を得たい。そこで HITS アルゴリズムに対してページの内容を閲覧して改善する手法として既存のテキスト解析手法とは違った Kolmogorov 記述量を用いた別の改善手法を提案し、その有効性を検討する。

1 はじめに

近年のインターネットの普及に伴い、World Wide Web は、情報を入手する目的で多くの人々に利用されている。この Web 上の情報はコンピュータネットワークの発達と共に増え続け、膨大な量の情報の中から有益な情報を見つけることは利用者にとって困難になるばかりである。そこで、このような状況の中で、有益な情報を得る為の方法が多く試みられている。その 1 つに、トピックを入力することで共通のトピックを持った Web ページ集合“ Web コミュニティ ”を抽出し、そのリンク構造に注目し、有益な情報を得る HITS アルゴリズム [1] がある。

この HITS はページ間のリンク構造だけを利用して Web ページの重み付けを行っており、各 Web ページの内容に立ち入らないことが前提となっている。つまり、探したいトピックに関係のないページも関係のあるページと同等に扱ってしまう。

そこで本研究では、Web ページの内容を考慮した重み付けを行うため、Kolmogorov 記述量を用いてページ間の類似度求める手法を提案し、有効性を評価するために、西村 [5] による Web ページのタグを利用したテキスト解析を用いた既存手法などを使って比較実験を行った。ちなみに Kolmogorov 記述量とはおおまかには、そのデータを作成するプログラムのうちサイズの最小のものである。以下、第 2 節で HITS アルゴリズムの手法を紹介し、第 3 節では HITS アルゴリズムの問題点、及び改善手法を記し、第 4 節で比較実験、評価結果を述べ、まとめとする

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に立ち入らずページ間のリンク構造を解析することにより、検索トピックに対して適切な情報 (“ Authority ”と “ Hub ”の 2 タイプからなる Web コミュニティ) を抽出するところにある。HITS アルゴリズムにおける “ Authority ”とは、特定のトピックにおける的確な情報を持ち、他のページからリンクされているページ群を意味する。また “ Hub ”とは、多くの Authority へリンクを張っているリンク集的なページ群である。

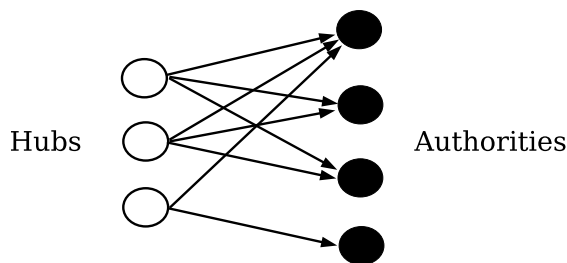


図 1: Hub と Authority の関係

2.2 HITS アルゴリズムの手順

STEP1

探したいトピックをサーチエンジンに投入し、一定

数 r 件の Web ページを収集し、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すべてのページ、及び root set に含まれるページにリンクしているページを 1 件につき最大 d 件収集し、root set に追加して大きさ n の base set を作成する。

STEP3

base set のすべてのページに対して、Authority と Hub の重みを付けていく。

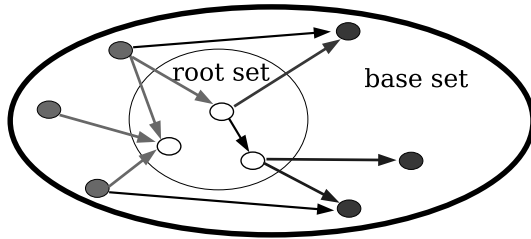


図 2: root set と base set の関係

2.2.1 Authority と Hub の重み付け

Authority と Hub は、各ページ p に与えられた Authority weight x_p と Hub weight y_p を計算する次の 2 式を反復することにより抽出される。

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

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

つまり、両者の関係は「よい Authority は複数の良質の Hub によってリンクされ、また、良質の Hub は複数のよいの Authority にリンクを張っている」と再帰的に定義している。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

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix}$$

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

反復計算

$Z = (1, 1, \dots, 1) \in R^n$ とし、 $X_0 = Z, Y_0 = Z$ とする。

そして更新ルールを

$$\begin{cases} X_i = A^T Y_{i-1} \\ Y_i = A X_i \end{cases}$$

として、極限の 2 式の操作を正規化を繰り返し行い、 X また、上記の 2 式より、反復回数を k とし、 $i = (1, 2, \dots, k)$ に対して

$$\begin{cases} X_k = (A^T A)^{k-1} A^T Z \\ Y_k = (A A^T)^k Z \end{cases}$$

の 2 式が導かれる。 $A^T A, A A^T$ は $n \times n$ の対称行列であるため、 X^* は $A^T A$ の最大固有値に対応する固有ベクトル、 Y^* は $A A^T$ の最大固有値に対応する固有ベクトルである。

なお、Kleinberg は反復回数を $k = 20$ と設定し、十分な結果を得ることが出来るとしている。

3 HITS アルゴリズムの改善

この章では、HITS アルゴリズムの問題点を挙げ、改善手法を提案する。

3.1 HITS アルゴリズムの問題点

HITS アルゴリズムは、各ページからのリンクは人為的に関係付けられたものであり重要な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。つまり、内容の異なったページにリンク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 に検索トピックとは無関係な、あまり適切では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関係のないページの weight が高くなってしまふ。これが topic drift 問題である。

3.2 改善の概要 (Kolmogorov 記述量)

以下に Kolmogorov 記述量を利用した改善手法をあげる。

各ページ間の類似度を Kolmogorov 記述量を用いて求め、それを隣接行列に反映させる。

3.2.1 類似度

各ページの Kolmogorov 記述量を用い、以下の式より類似度を求める。

$k(y) > k(x)$ のとき

$$d(x, y) = \frac{K(xy) - K(x)}{K(y)}$$

$*k(i)$ はページ i の Kolmogorov 記述量

* $k(xy)$ はページ xy を連結させたときの Kolmogorov 記述量

しかし Kolmogorov 記述量は計算不可能であるため、実際には近似値を用いて計算する。
近似値として圧縮したファイルのサイズを用いることができるので、以下のように式を変える。

$$d(x, y) = \frac{\text{zip}(xy) - \text{zip}(x)}{\text{zip}(y)}$$

* $\text{zip}(i)$ はページ i を圧縮した時のファイルサイズ

* $\text{zip}(xy)$ はページ x と y を連結させて圧縮した時のファイルサイズ

3.2.2 ページ間の隣接行列の改善

2.2.1 で示した隣接行列を次のように改善する。

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix}$$

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

* X には

$$1/d(x, y)$$

$$1 - d(x, y)$$

上記の式を使い、この式によって結果が変わることが予想される。

3.3 改善の概要 (既存手法)

以下に西村 [5] が提案した HTML タグを利用した改善手法を 2 つあげる。

(1) tag weight の導入

ページの内容に立ち入り、探したいトピックが特定の HTML タグで強調されているならば、その出現回数を隣接行列に反映させる。本研究では、特定の HTML タグとして“ $\langle B \rangle$ ” “ $\langle I \rangle$ ” “ $\langle META \rangle$ ” “ $\langle STRONG \rangle$ ” “ $\langle TITLE \rangle$ ” “ $\langle Hx \rangle$ ($x = 1, 2, 3, 4, 5$)”を使用した。

(2) anchor weight の導入

ページの内容に立ち入り、ページ内のアンカーテキスト、アンカータグ、及びアンカータグ前後のテキストに探したいトピックがあれば、その出現

回数を隣接行列に反映させる。ここでアンカータグは

“ $\langle A \text{ HREF} = \text{"http : //www..."} \rangle \sim \langle /A \rangle$ ” である。

(1) tag weight の導入は、ページ内で探したいトピックが HTML タグによって強調されているということは、そのページはトピックと関係していると思われ、ページの weight を重くしても良いと考えられるからである。しかし、Web ページは個人的なものも多く、開始タグのみで終了タグのない場合や、タグ本来の意味とは異なった意味で使われている場合など、きちんとしたタグの使い方がなされていない場合も多々ある。

そこで (2) anchor weight の導入では、ページ内のアンカーテキスト、アンカータグ、及びアンカータグ前後のテキストにはリンク先のページに関する説明が含まれている可能性が高いと考えられるので、この手法を提案する。

3.3.1 トピック値

前項の (1), (2) に当てはまる場合、ページに一定の値 (トピック値) を与える。

3.3.2 トピック値の決定

トピック w に対し、ページ p における w の出現回数を w_p とし、トピックベクトル W を作成する。

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

w_p = ページ p におけるトピック値

3.3.3 ページ間の隣接行列の改善

2.2.1 で示したページ間の隣接行列を次のように改善する。

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix}$$

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

以降は 2.2.1 と同様の手順で進める。

4 比較実験

4.1 比較方法

今回の実験では、いくつかのトピックに対し、以下の 10 個の方法を用いて比較実験を行った。

- ・比較対象
- (a) Google サーチエンジン [8]
- (b) HITS アルゴリズム
- (c) 既存手法 (tag weight)
- (d) 既存手法 (anchor weight)
- (e) HITS に Kolmogorov 記述量 ($1 - d(x, y)$) を導入したもの
- (f) HITS に Kolmogorov 記述量 ($1/d(x, y)$) を導入したもの
- (g) 既存手法 (tag) に kolmogorov 記述量 ($1 - d(x, y)$) を導入したもの
- (h) 既存手法 (anchor) に kolmogorov 記述量 ($1 - d(x, y)$) を導入したもの
- (i) Yahoo サーチエンジン [10]
- (j) HITS アルゴリズム 2

root set の r と d をそれぞれ $r = 100, d = 50$ に設定し、上記の (b)(c)(d)(e)(f)(g)(h) の 7 つのアルゴリズムにおいては、root set の収集は google Web API を用いて、
(j) の HITS アルゴリズム 2 においては Yahoo API を用いて行う。

4.1.1 web API

今現在ネットワークサービス企業が非商業目的利用を条件に、API を無料で公開しているところが増えてきている。

有名な企業では、
google
Yahoo!
MSN
AltaVista
などがあげられる。

例えば Google Web API [8] とは、開発者が自分専用のインターフェイスをつくり、Google の豊富で膨大な量のデータベースを利用して検索結果を好きなように利用することが出来る API のことである。また、API とは Application Programming Interface の略で、特定の

資源やアプリケーション、この場合は Google のデータベースにプログラムを使ってアクセスする際の入口のようなものである。

4.2 評価方法

評価方法として、アンケート方式で行った。調査対象者として、時間の関係上、谷研究室 21 名を対象にアンケート調査を行った。

アンケートの方法は、20 個のトピックにおいて (a)(b)(c)(d)(e)(f)(g)(h)(i)(j) の 10 個の方法でそれぞれの結果の上位 3 つの web ページを閲覧してもらい、内容を 3 段階で評価してもらった。

またどの方法がどの web ページを返したかは伏せて行った。

4.3 実験結果

10 個の方法で比較した結果、総合的に次のような支持率になった。

- (a) Google サーチエンジン : 13.78%
- (b) HITS アルゴリズム : 7.9%
- (c) 既存手法 (tag weight) : 9.59%
- (d) 既存手法 (anchor weight) : 10.14%
- (e) HITS に Kolmogorov 記述量 ($1 - d(x, y)$) を導入したもの : 9.2%
- (f) HITS に Kolmogorov 記述量 ($1/d(x, y)$) を導入したもの : 8.39%
- (g) 既存手法 (tag) に kolmogorov 記述量 ($1 - d(x, y)$) を導入したもの : 10.57%
- (h) 既存手法 (anchor) に kolmogorov 記述量 ($1 - d(x, y)$) を導入したもの : 11.08%
- (i) Yahoo サーチエンジン : 11.18%
- (j) HITS アルゴリズム 2 : 7.5%

具体的な結果の 1 部を最後に掲載。

4.4 考察

今回新たに、Kolmogorov 記述量を用いた改善手法を 2 つ、タグを利用した既存の改善手法に Kolmogorov 記述量を用いた改善手法を組み合わせた改善手法を 2 つ提案し、今回用いたトピックにおいて、多数に関して (HITS アルゴリズムに対して) 高い評価を受けることができた。

また、タグを利用した既存の改善手法に Kolmogorov 記述量を用いた改善手法を組み合わせることで、サーチエンジン以外では一番高い評価を受けることができた。

しかし、Kolmogorov 記述量を用いた改善手法では、タグを利用した既存の改善手法と比べて高い評価を受

けることができず、より良い有効性を示すことはできなかった。

このことから

Web ページの内容に立ち入って、ページ間の類似度を用いてページを評価することは有効であるが、既存手法など各ページのトピックとの関連性を用いる手法と組み合わせることで、より良い有効性を得られる。

次に、今回各アルゴリズムの root set の収集方法として主に Google Web API を用いたが、比較対象として Yahoo API を収集方法とした HITS アルゴリズム 2 を用意したが、HITS アルゴリズムと HITS アルゴリズム 2 の支持率にはほとんど違いはなかった。

このことから

HITS アルゴリズムにおいて、root set の収集方法の違いは有効性において影響は少ない。

最後に、予想どおり Kolmogorov 記述量を用いた改善手法において、類似度を使った式の違う 2 つの比較対象の支持率に違いがあった。

このことから

Kolmogorov 記述量を用いた改善手法において、類似度を使った式は結果に対して重要な影響をもつ。

といったことが得られた。

5 今後の課題

今回タグを利用した既存の改善手法に、Kolmogorov 記述量を用いた改善手法を組み合わせた際、全てのページに対して適用したが、ある一定以上のトピックの出現回数のあるページに絞ることで、より良い結果を得ることができるかもしれない。

また root set の収集方法として 2 つ (主には Google

Web API) の API を用いその結果、収集方法の違いによる支持率の違いはほぼ見られなかったが、他の (MSN, AltaVista など) API を用いることで結果・評価・支持率に違いがでてくるかもしれない。また改善手法においても、API を変えることで本実験と違った結果になるかもしれない。

最後に Kolmogorov 記述量を用いた改善手法での類似度を使った式を、今回は 2 種類用いたが他の式に変える事でより良い結果がでてくるかもしれない。

参考文献

- [1] John M.Kleinberg, “Authoritative Sources in a Hyperlinked Enviroment,” Research Report RJ 10076, IBM. 1997.
- [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] 武田 善行, 梅村 恭司, 藤井 敦, Web マイニング, 共立出版, 2004.
- [4] 谷 聖一 データの複雑さ・データ表現の複雑さ
- [5] 西村 有美子 半構造データを用いた WWW リンク構造解析アルゴリズムの新提案, 2005
- [6] 三井 貴子 Kolmogorov Complexity を用いた民族音楽分類, 2005
- [7] Google, <http://www.google.co.jp/>
- [8] Google Web API, <http://www.google.com/apis/>
- [9] Jama, <http://math.nist.gov/javanumerics/jama/>
- [10] Yahoo Developer Network, <http://developer.yahoo.net/>

実験結果

(1) 既存手法 (Tag) に Kolmogorov 記述量の導入したのが一番良かったもの

topic : データベース

・ Google

順位	URL
1	http://macao.softvision.co.jp/dbpwww/
2	http://www.tvdrama-db.com/
3	http://www.wakhok.ac.jp/DB/DB.html

・ Yahoo!

1	http://us.imdb.com/
2	http://www.meganslaw.ca.gov/
3	http://www.linuxprinting.org/printer_list.cgi

・ HITS アルゴリズム

1	http://www.dnp.co.jp/artscape/index.html
2	http://www.dnp.co.jp/artscape/privacy/index.html
3	http://www.dnp.co.jp/artscape/site.html

・ 既存手法 (tag)

1	http://www.cric.or.jp/db/dbfront.html
2	http://w-chemdb.nies.go.jp/
3	http://www.dpc.or.jp/

・ 既存手法 (anchor)

1	http://www.ne.jp/asahi/coffee/house/DB/400.html
2	http://www.ne.jp/asahi/coffee/house/DB/300.html
3	http://www.ne.jp/asahi/coffee/house/DB/500.html

・ HITS に Kolmogorov 記述量 ($1-d(x,y)$) を導入したもの

1	http://www.trendmicro.com/jp/security/encyclopedia/
2	http://jmdb.club.ne.jp/
3	http://www1.river.go.jp/

・ HITS に Kolmogorov 記述量 ($1/d(x,y)$) を導入したもの

1	http://www.trendmicro.co.jp/infos/default.asp?advis=&sort=pattern&page=1
2	http://www.trendmicro.co.jp/infos/default.asp?advis=&sort=date&order=asc&page=1
3	http://www.trendmicro.co.jp/infos/default.asp?advis=more&sort=date&order=desc

・ 既存手法 (tag) に Kolmogorov 記述量 ($1-d(x,y)$) を導入したもの

1	http://www.cric.or.jp/db/dbfront.html
2	http://e-words.jp/w/E38387E383BCE382BFE38399E383BCE382B9.html
3	http://www.d4.dion.ne.jp/warapon/data00/

・ 既存手法 (anchor) に Kolmogorov 記述量 ($1-d(x,y)$) を導入したもの

1	http://e-words.jp/w/E38387E383BCE382BFE38399E383BCE382B9.html
2	http://www.ne.jp/asahi/coffee/house/DB/400.html
3	http://www.ne.jp/asahi/coffee/house/DB/500.html

・HITS アルゴリズム 2

1	<i>http://www.imdb.com/rg/nav – home/navbar/</i>
2	<i>http://www.imdb.com/rg/sub – home/navbar/</i>
3	<i>http://www.imdb.com/rg/sub – top/navbar/Top/</i>

(2) 既存手法 (Anchor) に Kolmogorov 記述量の導入したのが一番良かったもの

topic : cake

・ Google

順位	URL
1	http://www.cakemusic.com
2	http://www.cakemusic.com/music.html
3	http://cake.allrecipes.com/

・ Yahoo!

1	http://www.cakemusic.com/
2	http://cake.allrecipes.com/
3	http://www.sonymusic.com/artist/Cake/

・ HITS アルゴリズム

1	http://bbonline.com/index.html
2	http://bbonline.com/recipe/ponderosa_w_recipe1.html
3	http://bbonline.com/recipe/charm_a_recipe1.html

・ 既存手法 (tag)

1	http://www.cs.cmu.edu/~mjw/recipes/cake/mp-cake-coll-2.html
2	http://www.cs.cmu.edu/~mjw/recipes/cake/mp-cake-coll-3.html
3	http://www.cs.cmu.edu/~mjw/recipes/cake/mp-cake-coll-4.html

・ 既存手法 (anchor)

1	http://www.astray.com/recipes/?search=Cakes
2	http://www.astray.com/recipes/?show=8-Minute%20cheesecake
3	http://www.astray.com/recipes/?show=8-Minute%20chicken%20pie

・ HITS に Kolmogorov 記述量 ($1-d(x,y)$) を導入したもの

1	http://southernfood.about.com/cs/appetizers/a/party_snacks.htm
2	http://southernfood.about.com/cs/breadmachinerecipes/tp/aatpbread1.htm
3	http://southernfood.about.com/cs/crockpotrecipes/tp/tpcrockpots.htm

・ HITS に Kolmogorov 記述量 ($1/d(x,y)$) を導入したもの

1	http://en.wikipedia.org/w/index.php?title=Cake%28band%29&action=edit&section=1
2	http://en.wikipedia.org/w/index.php?title=Cake%28band%29&action=edit&section=2
3	http://en.wikipedia.org/w/index.php?title=Cake%28band%29&action=edit&section=3

・ 既存手法 (tag) に Kolmogorov 記述量 ($1-d(x,y)$) を導入したもの

1	http://www.bakingshop.com/index.html
2	http://www.bakingshop.com/weddingcaketoppers/preciousmoments.htm
3	http://www.bakingshop.com/weddingcaketoppers/hispanic.htm

・ 既存手法 (anchor) に Kolmogorov 記述量 ($1-d(x,y)$) を導入したもの

1	http://www.cakepia.info/cooking/search/detail.html?LMID=707
2	http://www.cakepia.info/cooking/search/detail.html?LMID=708
3	http://www.cakepia.info/cooking/search/detail.html?LMID=706

・HITS アルゴリズム 2

1	<i>http://www.imdb.com/rg/nav-home/navbar/</i>
2	<i>http://www.imdb.com/rg/sub-home/navbar/</i>
3	<i>http://www.imdb.com/rg/sub-top/navbar/Top/</i>