Kolmogorov 記述量に基づく類似度を用いたWWW リンク構造解析手法

真板 育海
2/10

目次

- 目的
- HITS アルゴリズムの紹介
- 改善案の紹介
- 比較実験
- 考察

目次

目的

- コンピュータネットワークの発達とともに情報が溢れている
- そのような状況の中で自分の探したいトピックに関係している web ページを探したい

目的

- 入力：探したいトピック

WWW リンク構造を解析（HITS アルゴリズム）

出力：関連性の高い Web ページ

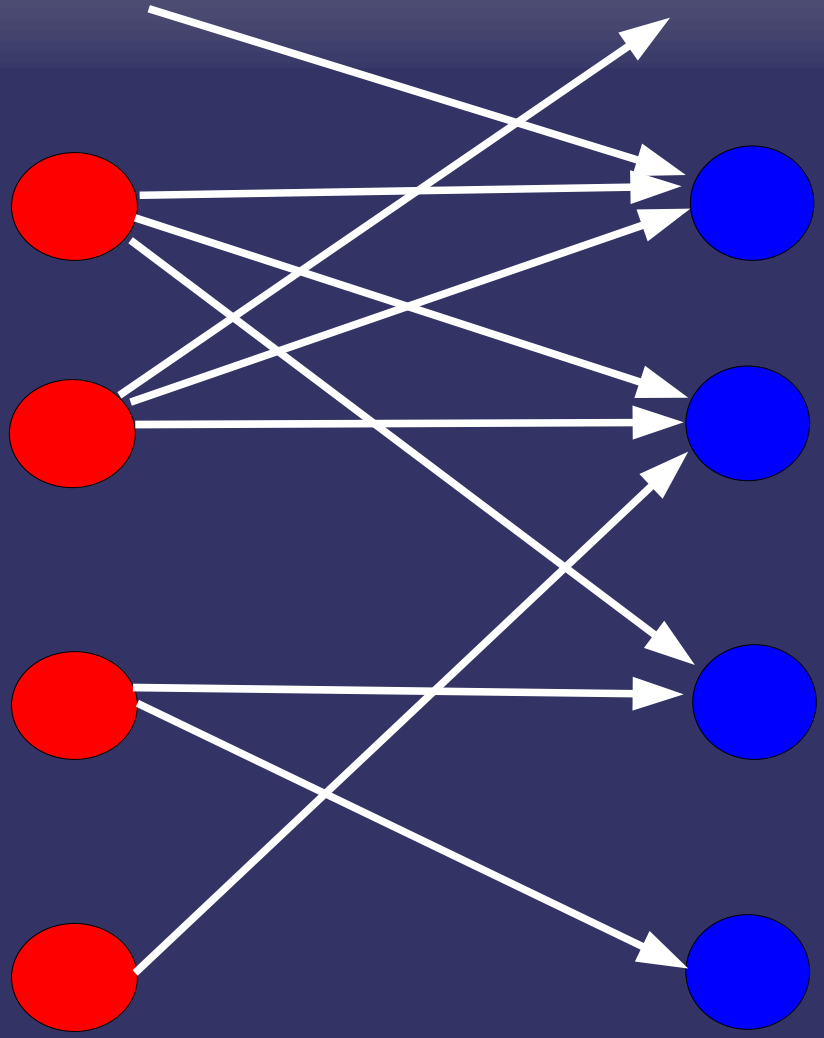
HITS アルゴリズム

- 特徴：ページの内容に立ち入らずに
  ページ間のリンク構造のみに着目して
  トピックに関連した
  Authority や Hub の
  ページ集合を抽出

- Jon M.Kleinberg

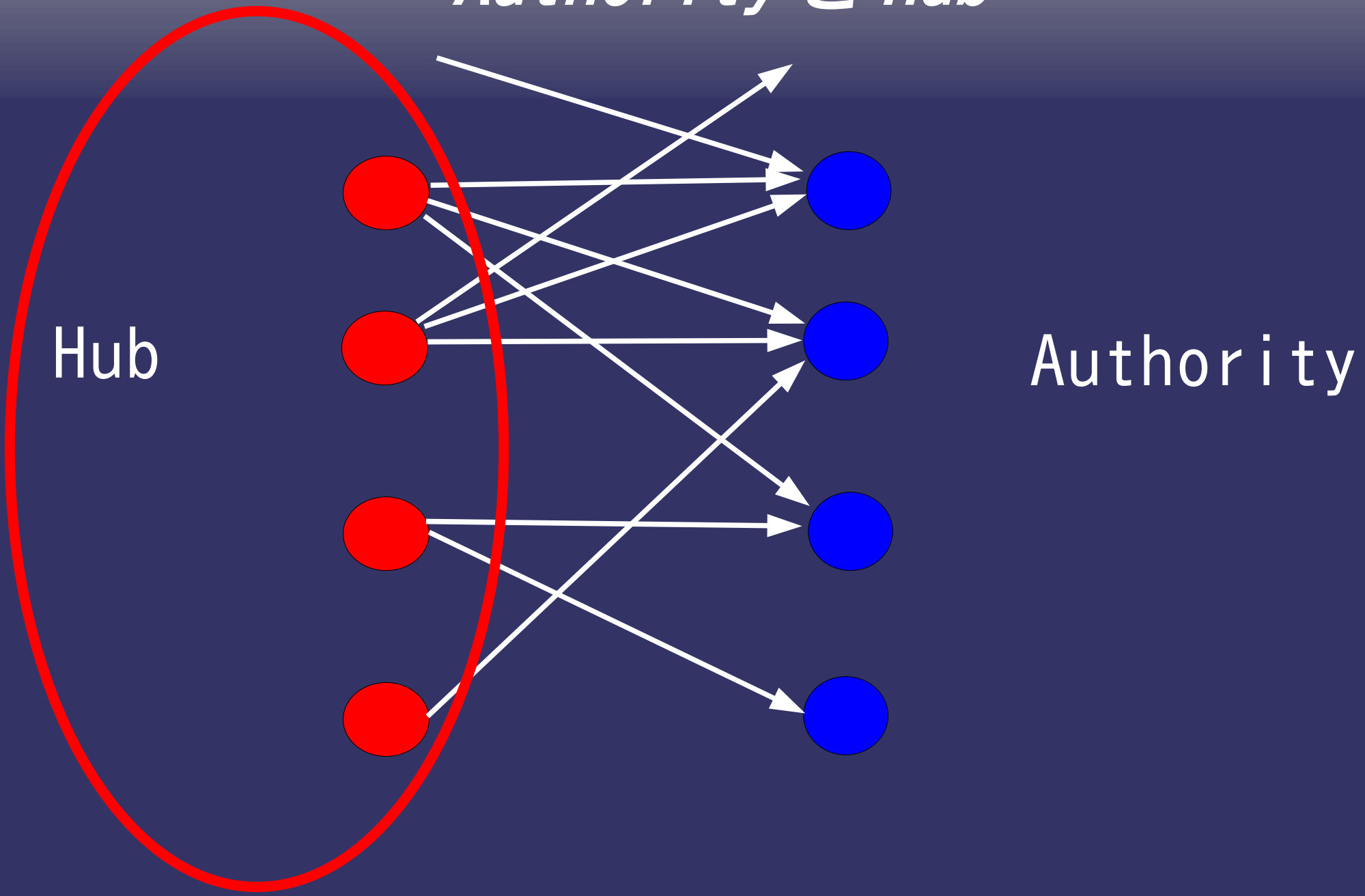
Authority と Hub

Hub

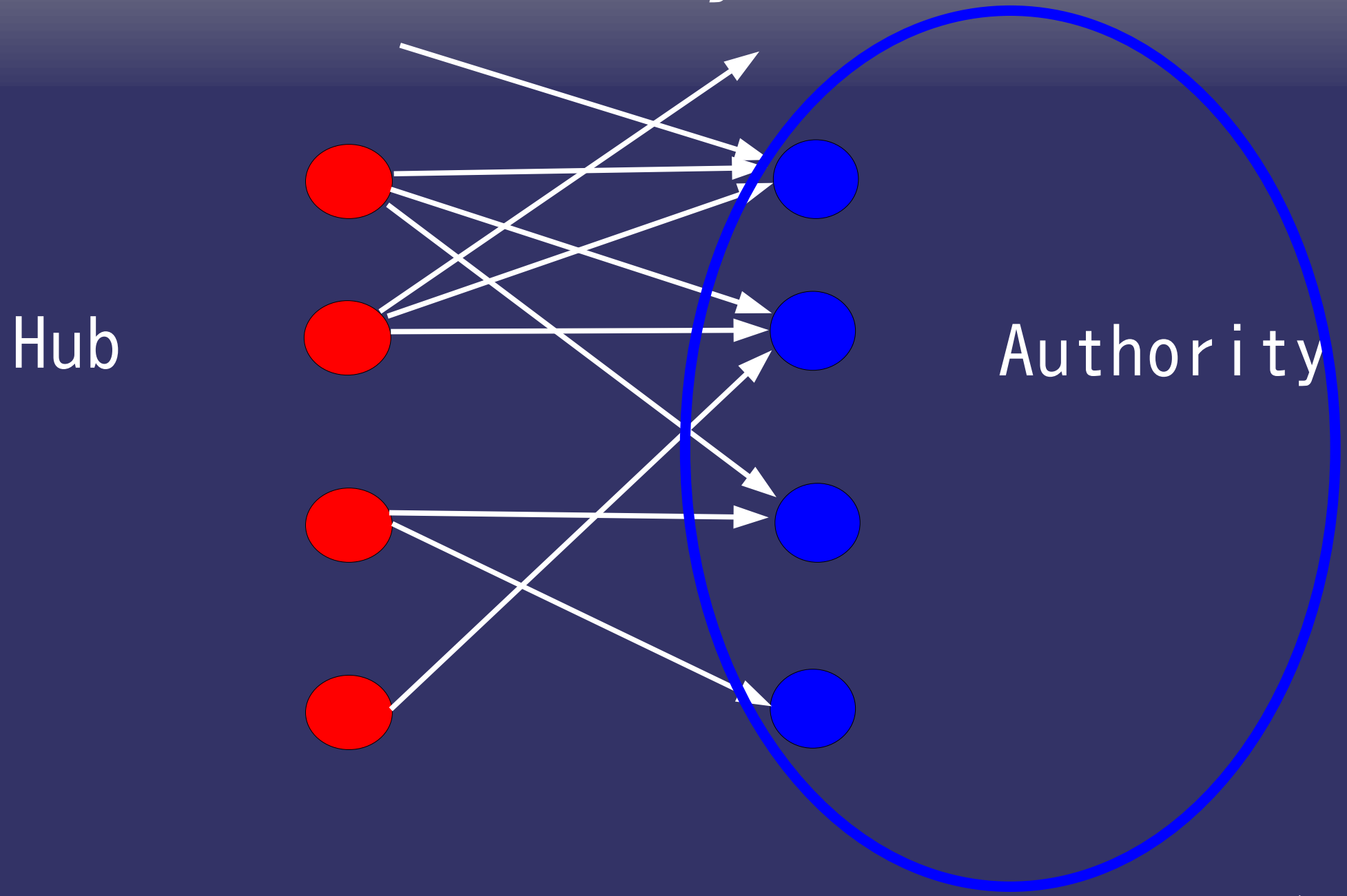


Authority

Authority と Hub



Authority と Hub



目的

- さらにページ内容に立ち入り
  ページ間の類似性という観点から
  Kolmogorov 記述量に基づく類似度を使い
  HITS アルゴリズムの改善手法を提案

既存手法（西村）との比較実験を行う

目次

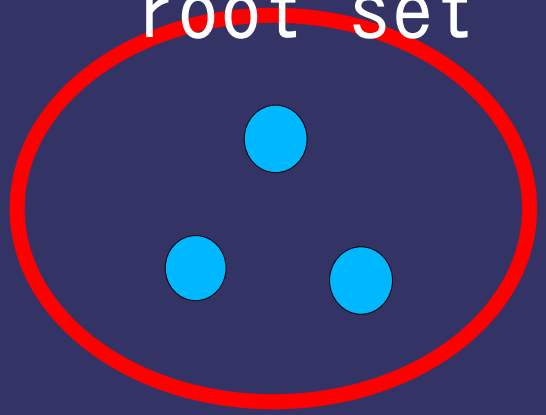
- 目的
- HITS アルゴリズムの紹介
- 改善案の紹介
- 比較実験
- 今後の課題

HITS アルゴリズムの手順

- １．root set の作成
- ２．base set の作成
- ３．Authority と Hub の重み付け

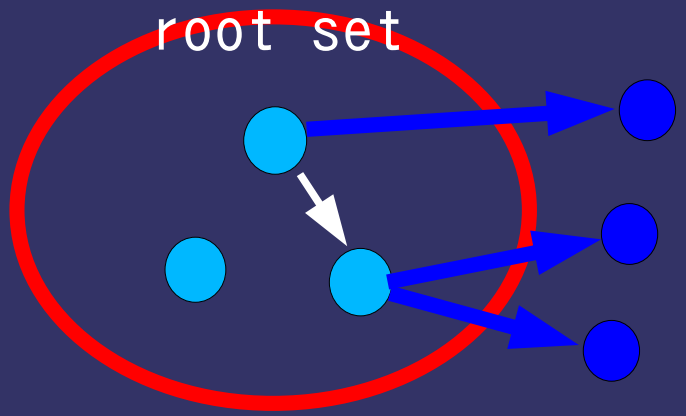
1、 root set の作成

- 探したいトピックを検索エンジンにかけ
  r 件の Web ページを収集し
  root set とする



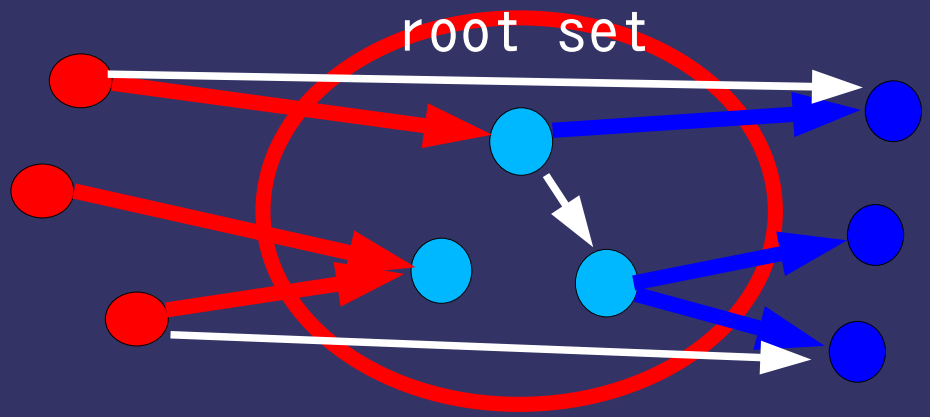
2、 base set の作成

- root set に含まれるページからリンクされているページを収集



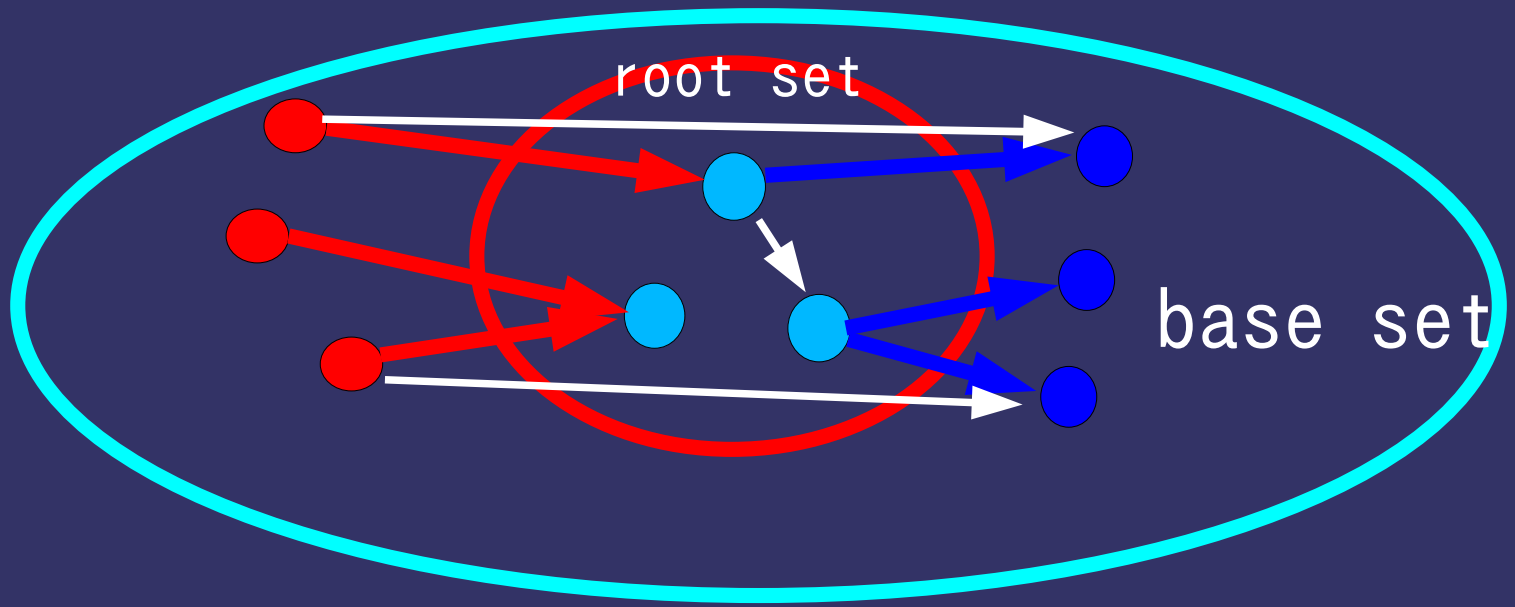
2、 base set の作成

- 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



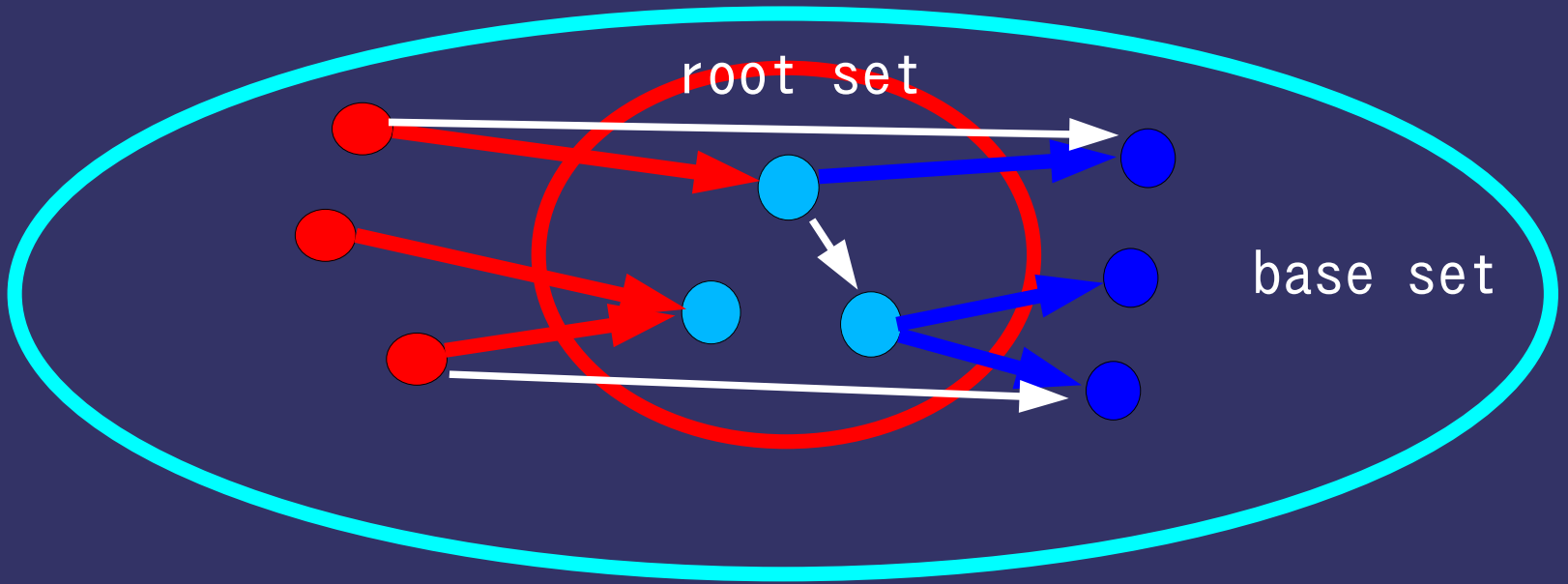
2、 base set の作成

- root set に含まれるページからリンクされているページ
  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）収集し
  root set に追加して base set を作成

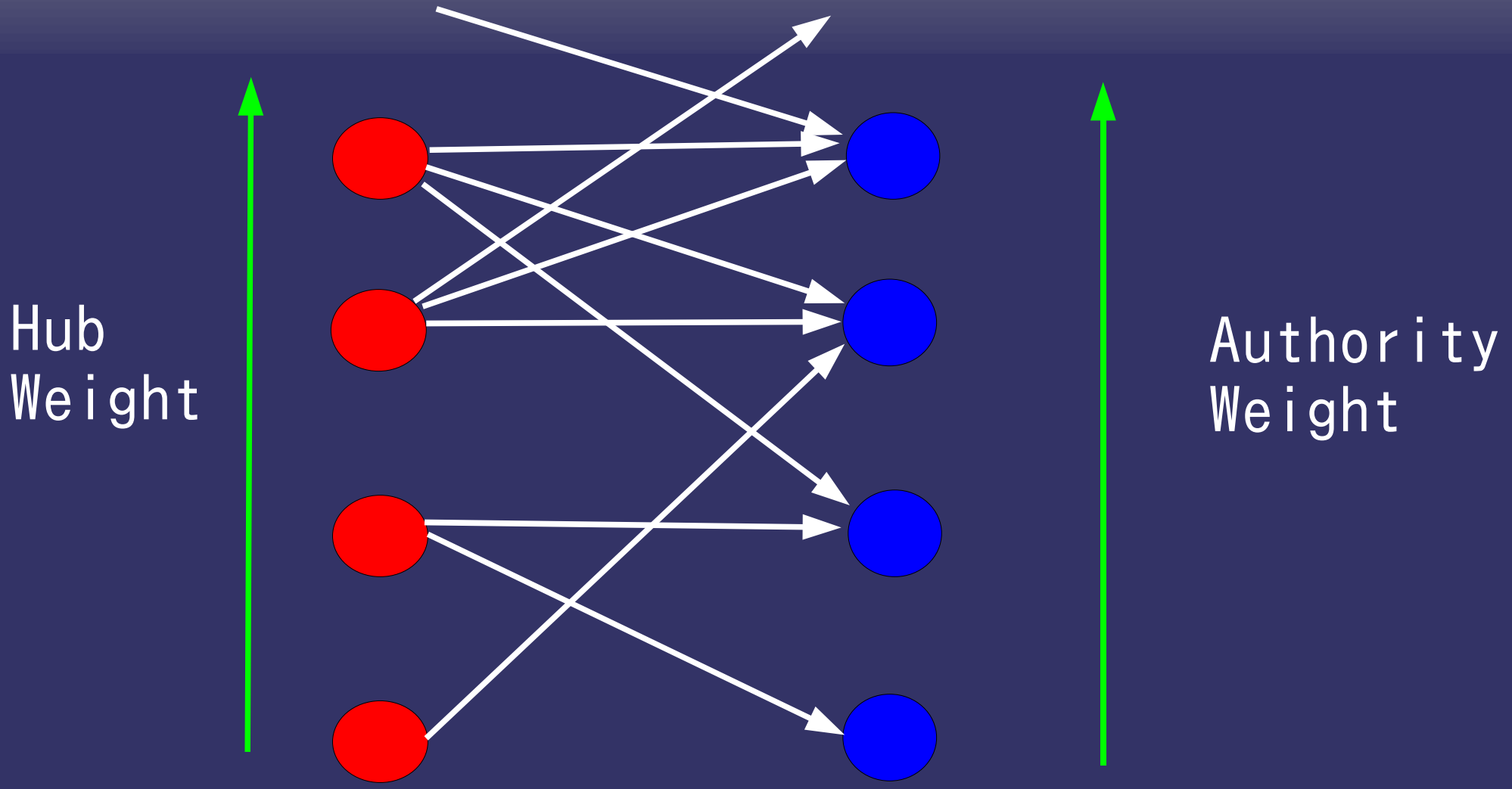


3、 Authority と Hub の重み付け

- base set のページについて
  Authority と Hub に重みを付けていく



Authority と Hub の重み



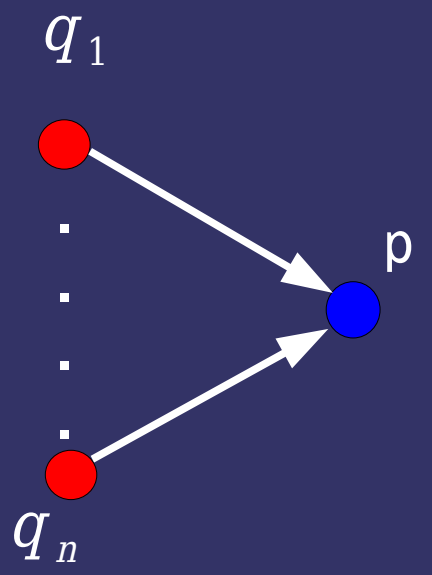
Authority と Hub の重み付け方

- ページ p があるとき
- $x(p)$: Authority weight
- $y(p)$: Hub weight

Authority と Hub の重み付け方

- Authority weight $x(p)$:

このようなとき
x(p) は p にリンクをはっている
全ての q の持つ y(q) の総和である



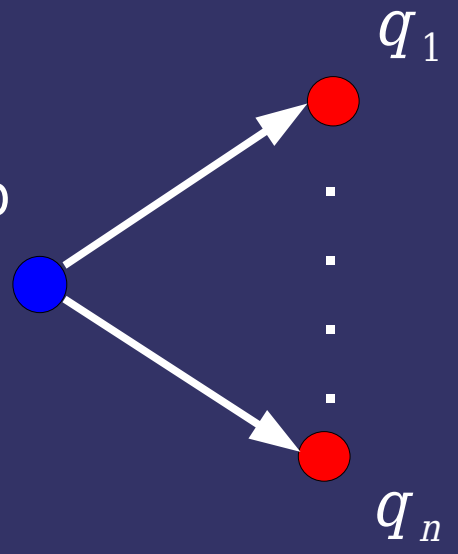
$$\rightarrow x(p) = y(q_1) + \ldots + y(q_n)$$

$$\sum_{i=1}^{n} y(q_i)$$

Authority と Hub の重み付け方

- Hub weight $y(p)$:

このようなとき
y(p) は p からリンクをはられている
全ての q の持つ x(q) の総和である



→ $y(p)=x(q_1)+\ldots+x(q_n)$

$$\sum_{i=1}^{n} x(q_i)$$

Authority と Hub の重み付け方

- 下記の式を反復することで
  それぞれの値を更新していく

- 最終的に Authority weight の高いページを
  抽出する

$$x(p)=\sum_{i=1}^{n} y(q_i) \qquad y(p)=\sum_{i=1}^{n} x(q_i)$$

反復計算

- リンク構造を隣接行列と考える
- 隣接行列　A:

$$A=\begin{bmatrix} a(1,1) & \cdots & a(1,n) \\ \vdots & & \vdots \\ a(n,1) & \cdots & a(n,n) \end{bmatrix}$$

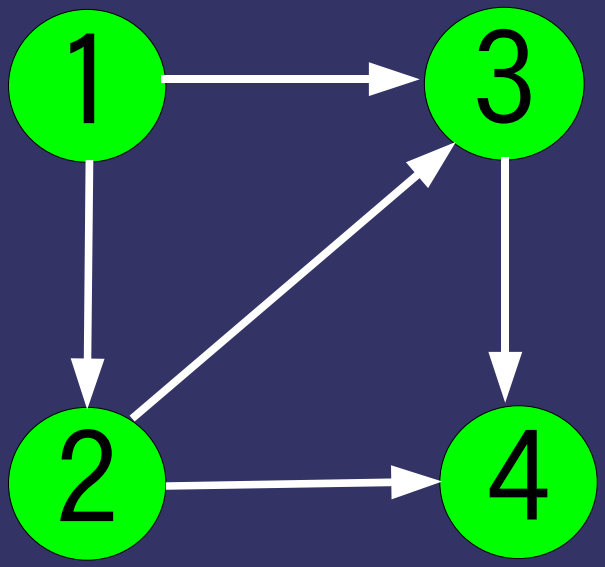
$$a(i,j)=\begin{cases} 1: i \to j \text{ が存在するとき} \\ 0: \text{しないとき} \end{cases}$$

反復計算

- リンク構造を隣接行列と考える
- 隣接行列　A:

$$A=\begin{bmatrix} a(1,1) & \cdots & a(1,n) \\ \vdots & & \vdots \\ a(n,1) & \cdots & a(n,n) \end{bmatrix}$$

$$a(i,j)=\begin{cases} 1: i \rightarrow j \text{ が存在するとき} \\ 0: \text{しないとき} \end{cases}$$

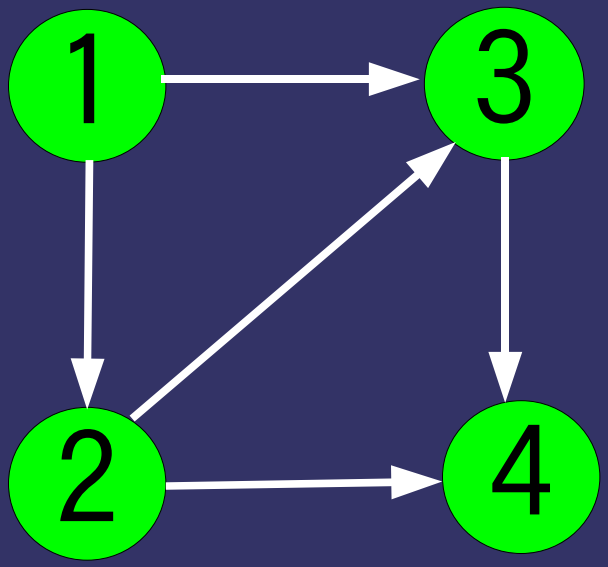


反復計算

- リンク構造を隣接行列と考える
- 隣接行列　A:

$$A=\begin{bmatrix} a(1,1) & \cdots & a(1,n) \\ \vdots & & \vdots \\ a(n,1) & \cdots & a(n,n) \end{bmatrix}$$

$$a(i,j)=\begin{cases} 1: i \rightarrow j \text{ が存在するとき} \\ 0: \text{しないとき} \end{cases}$$



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

計算方法

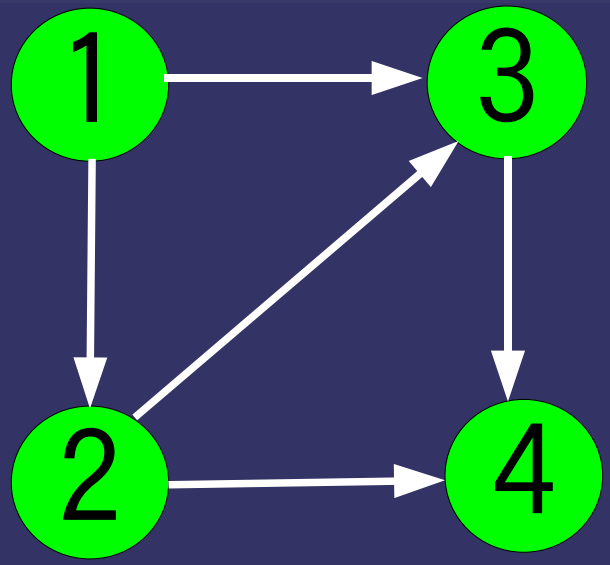
- A: 隣接行列
- $X_i=[x(1),\ldots,x(n)]$:Authority 用ベクトル
- $Y_i=[y(1),\ldots,y(n)]$:Hub 用ベクトル
- $Z=[1,\ldots,1]\in R^n$

- 初期値： $X_0=Z \quad Y_0=Z$

- 更新のルール： $X_i=A^T Y_{i-1}$
  $Y_i=A X_i$

  i：反復回数
  (i=1,2,...,k)

反復計算



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

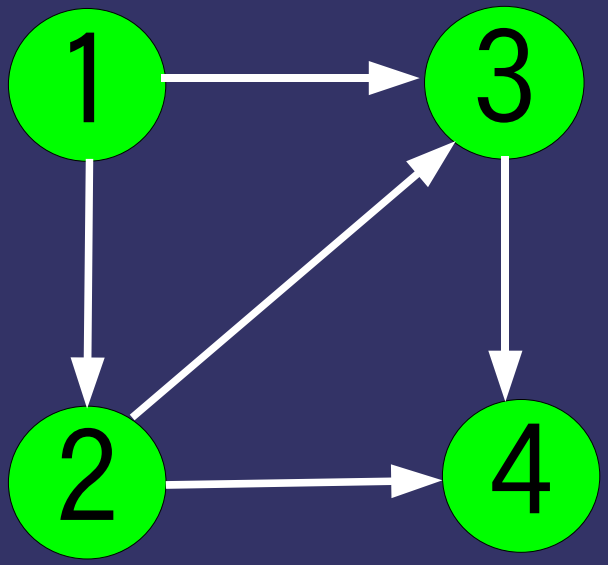
$$X_1 = A^T Y_0$$

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Authority Weight

Hub Weight

反復計算



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

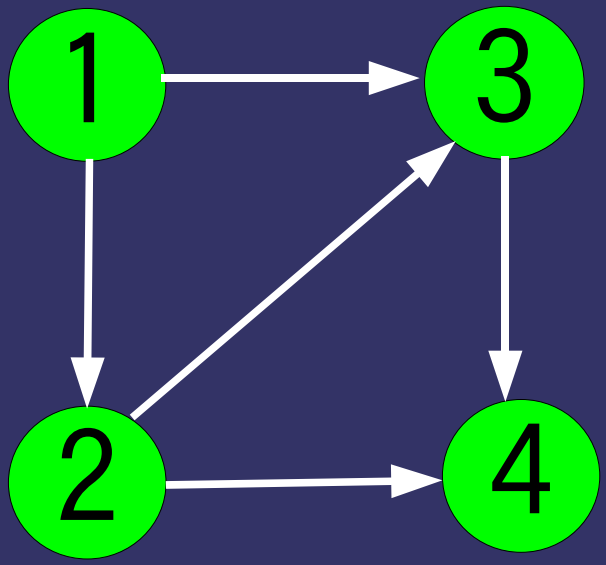
$$Y_1 = A X_1$$

$$\begin{bmatrix}3\\4\\2\\0\end{bmatrix} = \begin{bmatrix}0&1&1&0\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}\begin{bmatrix}0\\1\\2\\2\end{bmatrix}$$

Hub Weight

Authority Weight

反復計算



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

$$\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\ \begin{bmatrix}1\\1\\1\\1\end{bmatrix}\rightarrow\begin{bmatrix}0\\1\\2\\2\end{bmatrix}\ \begin{bmatrix}3\\4\\2\\0\end{bmatrix}\rightarrow\begin{bmatrix}0\\3\\7\\6\end{bmatrix}\ \begin{bmatrix}10\\13\\6\\0\end{bmatrix}\rightarrow\begin{bmatrix}0\\10\\23\\19\end{bmatrix}\ \begin{bmatrix}33\\42\\19\\0\end{bmatrix}$$

X_0 Y_0 X_1 Y_1 X_2 Y_2 X_3 Y_3

目次

- 目的
- HITS アルゴリズムの紹介
- 改善案の紹介
- 比較実験
- 考察

改善案（西村）

HITS は Web ページの内容に立ち入らない

→ トピックとは関係ないページを関係あるページと同じように扱ってしまう

トピックと関係あるページの Weight を重くしたい

ページの内容に立ち入り
Web ページのタグの部分について着目している

改善案（西村）

(1)tag weight

探したいトピックが特定のタグで強調
されているならば
その出現回数をリンクの重みに反映させる

＊特定のタグ

<TITLE>topic</TITLE>
<STRONG>topic</STRONG>
<H6>topic</H6>

.
.
.

など

改善案（西村）

(2)anchor weight

ページ内のアンカーテキスト
アンカータグ
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
その出現回数をリンクの重みに反映させる

＊例
....
topic
topicの概要

など

改善案（西村）

$a(i,j)=$

1: i→j が存在するとき

0: しないとき

$a(i,j)=$

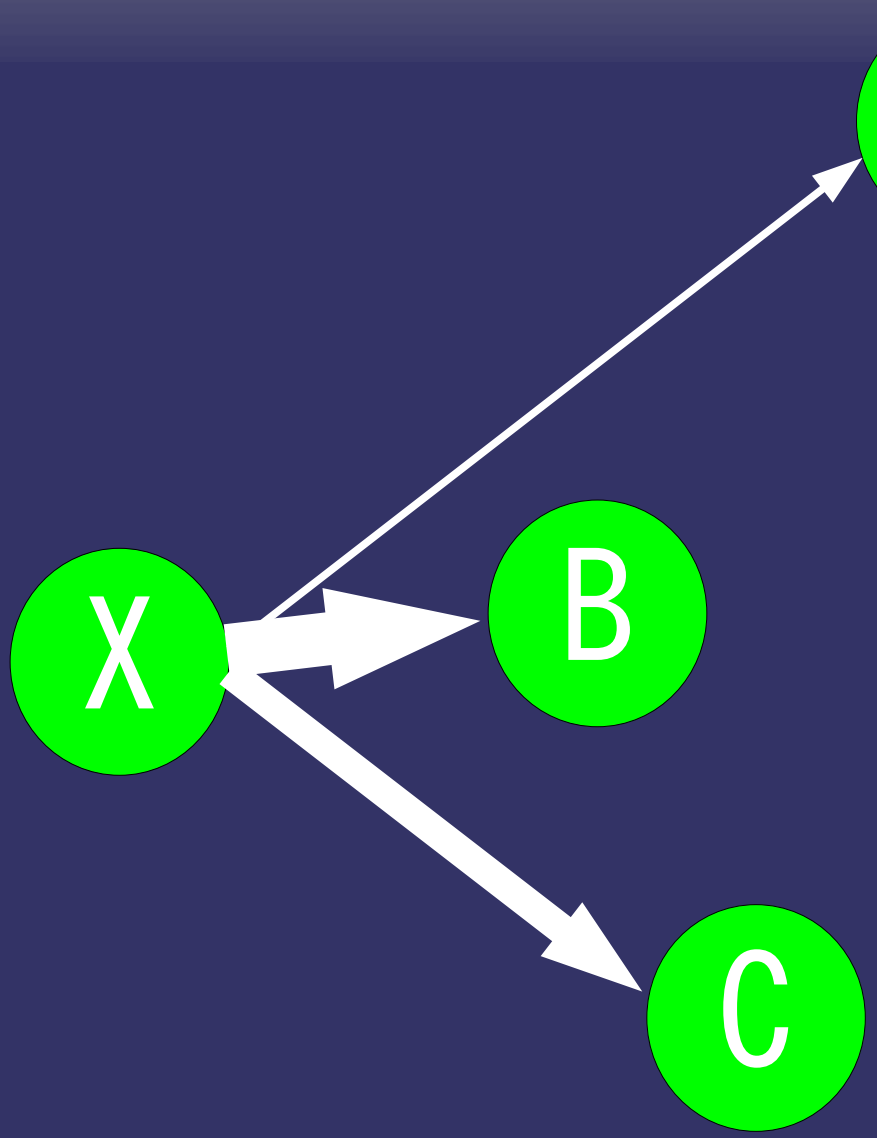
1 ＋出現回数: i→j が存在するとき

0 : しないとき

改善案(Kolmogorov)

- ページの内容に立ち入り
  各ページ間の類似性に着目した
  改善案を提案したい

類似性と振り分けの関係



Bとの類似性

＞Cとの類似性

＞Aとの類似性

Bへの振り分け

＞ Cへの振り分け

＞ Aへの振り分け

改善案(Kolmogorov)

- 元のページに対して
  類似しているページは
  元のページとの関連性が強い
  (同じ対象を扱っている可能性が高い)
  のではと考えられる

  元のページと類似性の高いページに対して
  多く重みを付けたい

改善案（Kolmogorov）

- Kolmogorov 記述量に基づき
  二つのデータの類似性を類似度として求めることができる

  リンクのある２つのページの類似度を求め
  それに応じた振り分けをリンクの重みに反映させる

  ＊対象としてタグ等を除いた本文のみとする

類似度

ページｉ，ｊの類似度をKolmogorov記述量に基づいて求める式は

$k(j)>k(i)$ のとき

$$d(i,j)=\frac{k(ij)-k(i)}{k(j)}$$

*k(i):iのKolmogorov記述量
*ij:iとｊを連結させたもの

類似度の近似

しかしKolmogorov記述量は計算不可能であることが知られている

近似値として圧縮した後のサイズを用いることができる

$$\frac{zip(ij)-zip(i)}{zip(j)}$$

*zip(i):iを圧縮した後のサイズ

改善案(Kolmogorov)

$$a(i,j)=\begin{cases}1: i\rightarrow j \text{ が存在するとき}\\ 0: \text{しないとき}\end{cases}$$

$$a(i,j)=\begin{cases}1-d(i,\ j) & : i\rightarrow j \text{ が存在するとき}\\ 0 & : \text{しないとき}\end{cases}$$

振り分け方

- 他に

$$\frac{1}{d(i,j)}$$

- また既存手法と組み合わせることで

$$(1+wj)*(1-d(i,j))$$

目次

- 目的
- HITS アルゴリズムの紹介
- 改善案の紹介
- 比較実験
- 考察

比較対象

- 1，Google　サーチエンジン

  2，HITS　アルゴリズム

  3，　既存手法（tag weight）

  4，　既存手法（anchor　weight）

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

比較対象

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

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

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

9, Yahoo! サーチエンジン

10, HITS アルゴリズム 2

比較方法

- root set の収集方法

  2 〜 8 : Google web API

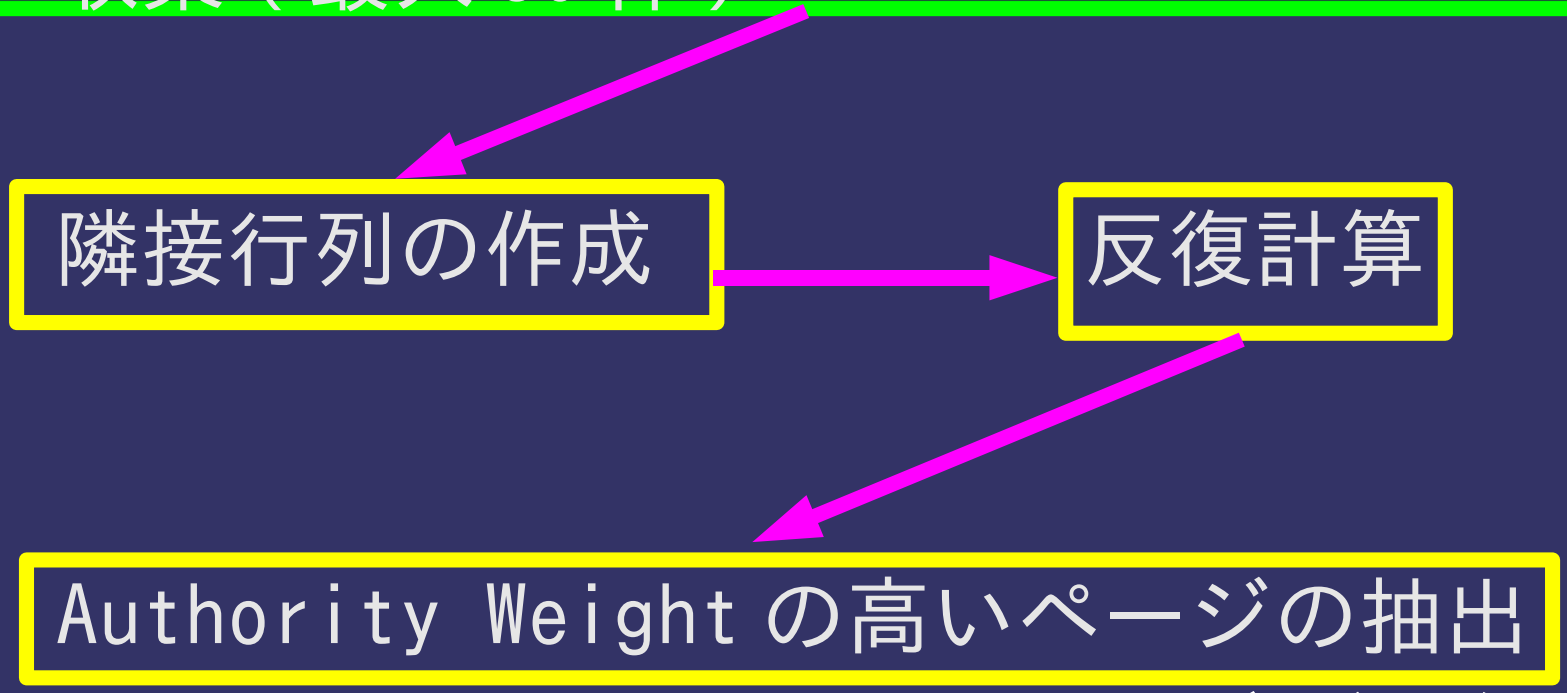
  10 : Yahoo! API

base set

- root set 収集（100 件）

  root set のページからリンクされている
  ページ収集

  root set のページにリンクしているページ
  収集（最大 50 件）



評価方法

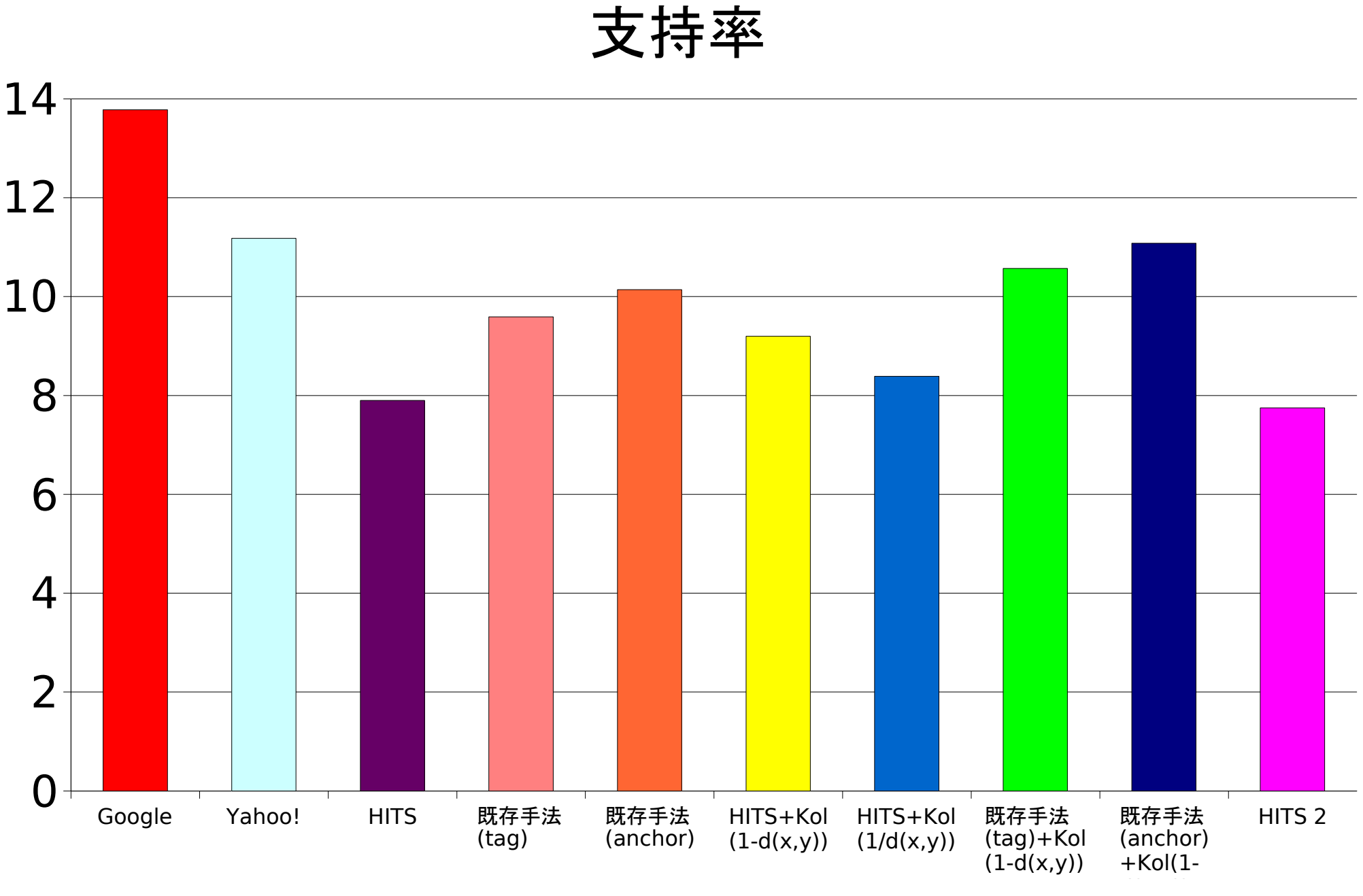
- 評価方法

  アンケート方式
- 対象者

  谷研究室　3.4 年　21 名
- アンケートの方法

  比較対象それぞれにトピックごとに
  結果の上位３位のページを閲覧し
  内容を３段階で評価してもらった
  どの比較対象がどのページを返したかは
  伏せて行った

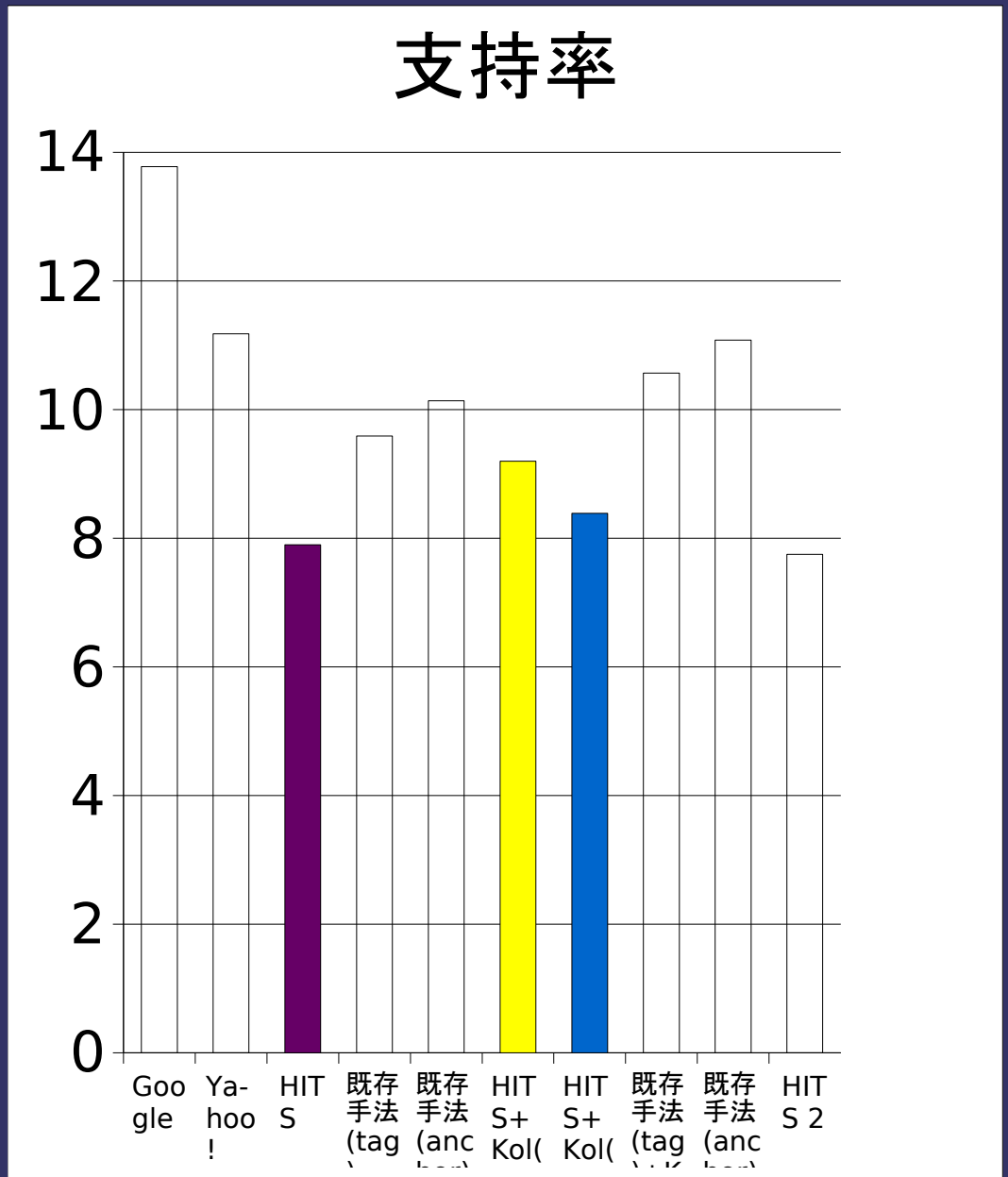
実験結果


支持率
14
12
10
8
6
4
2
0
Google
Yahoo!
HITS
既存手法 (tag)
既存手法 (anchor)
HITS+Kol (1-d(x,y))
HITS+Kol (1/d(x,y))
既存手法 (tag)+Kol (1-d(x,y))
既存手法 (anchor) +Kol(1-
HITS 2

目次

- 目的
- HITS アルゴリズムの紹介
- 改善案の紹介
- 比較実験
- 考察

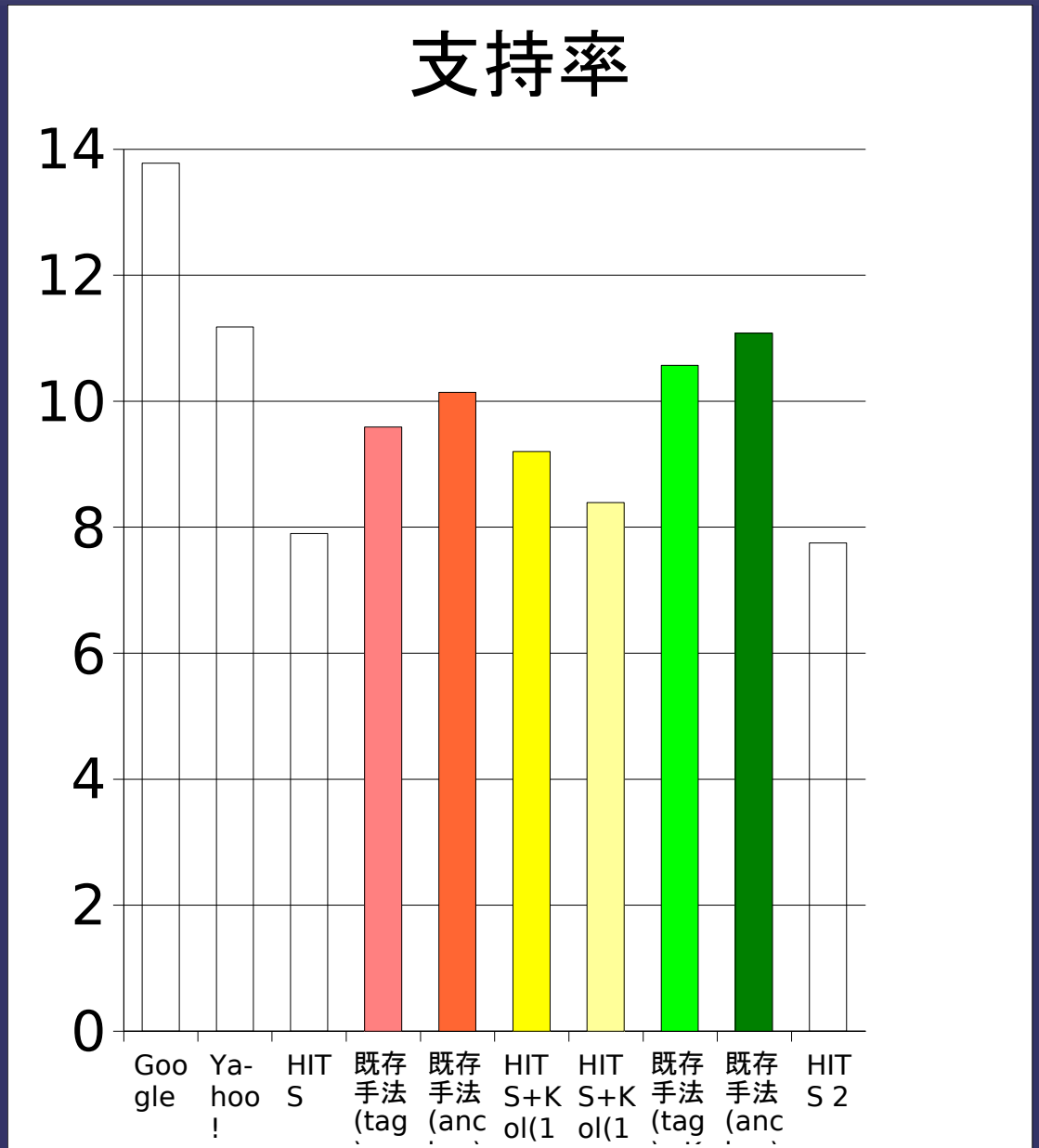
考察



- 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

有効である

考察



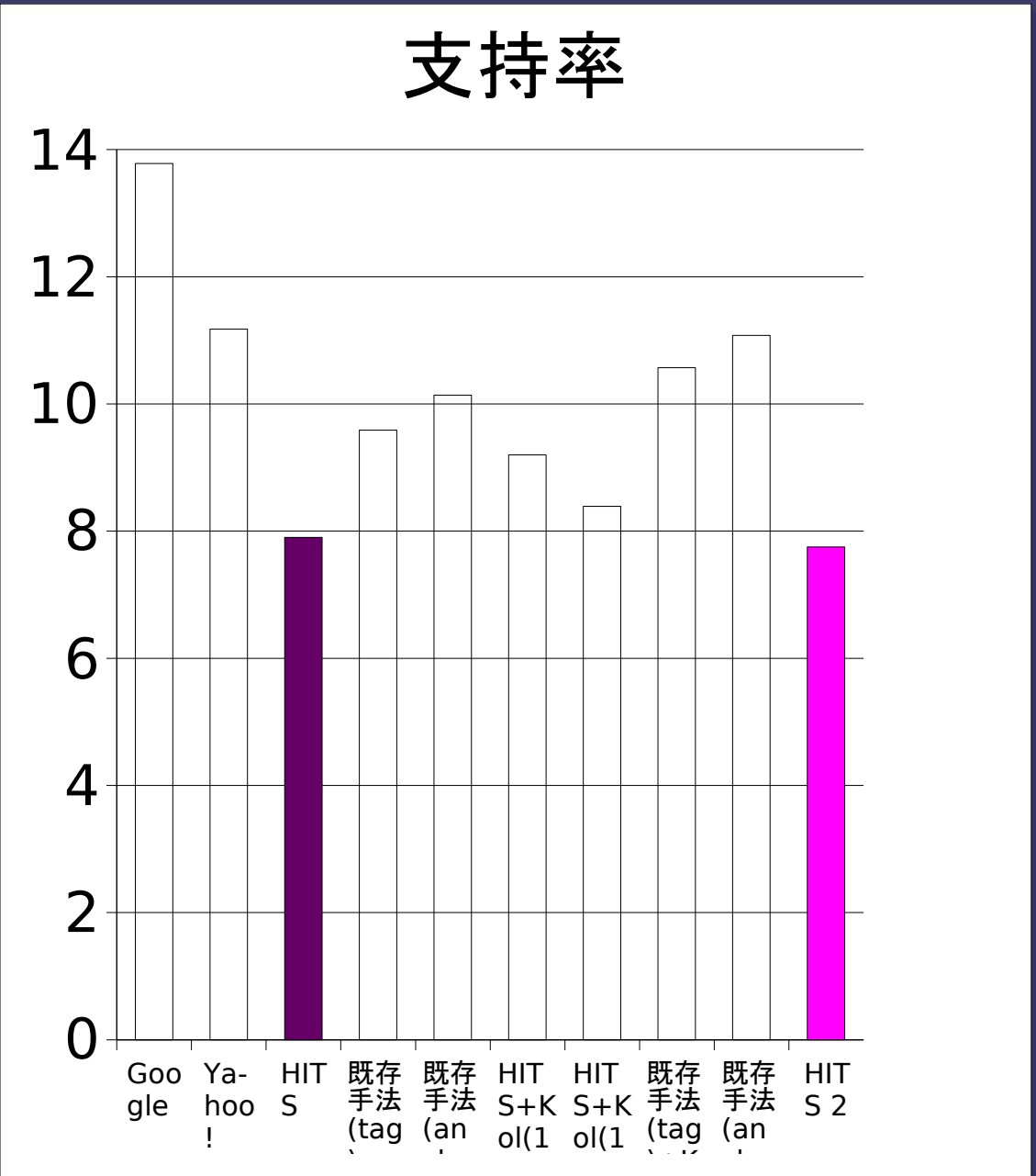
- Kolmogorov 記述量を用いた改善手法は

  他の手法と組み合わせた方が

  ↓

  さらに有効である

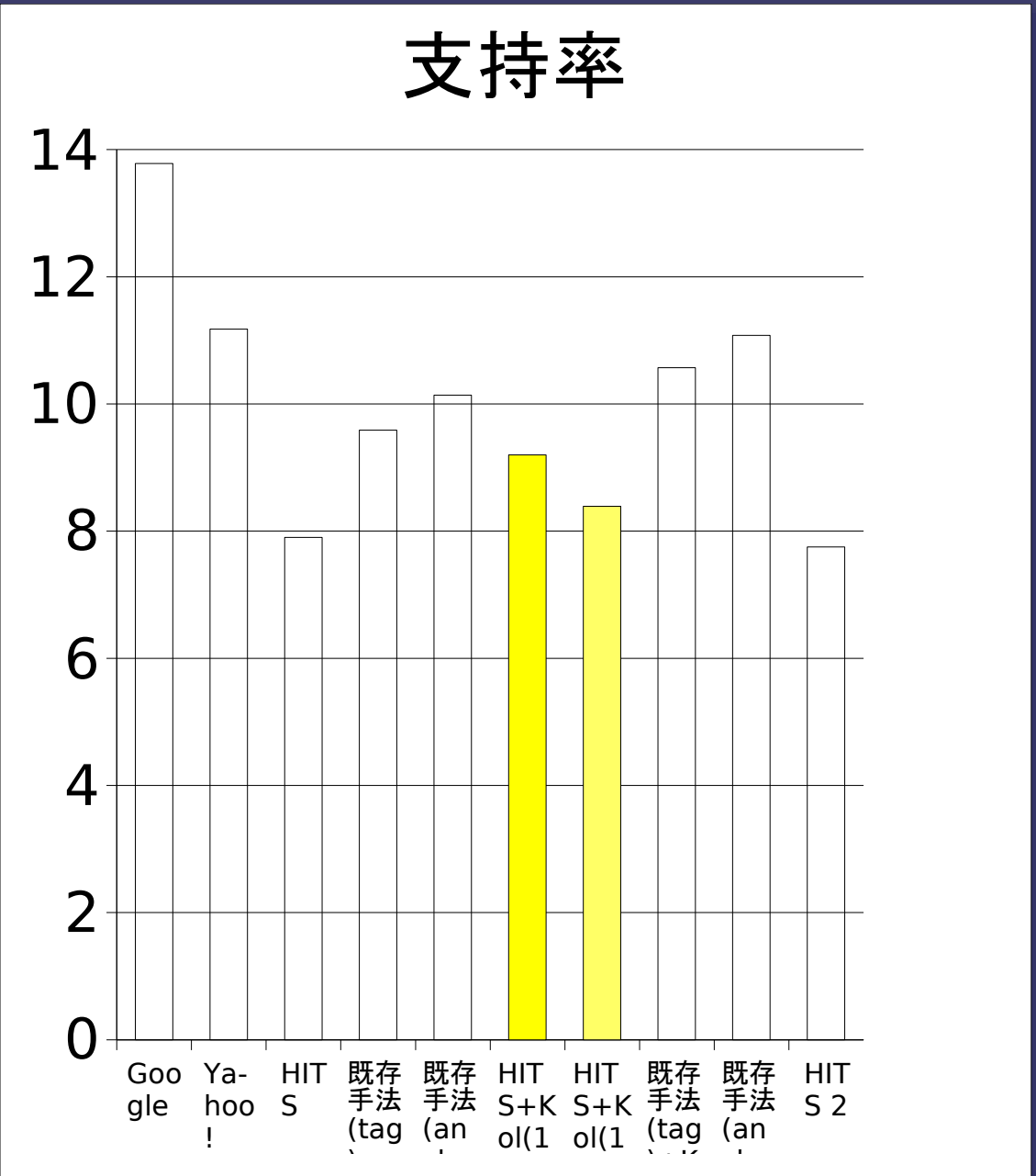
考察



- root set 収集方法の違い

有効性において影響は少ない

考察



- Kolmogorov を用いた改善手法において

  類似度による
  リンクの重み付け

  ↓

  評価に影響がある

今後の課題

- root set の収集方法

  今回用いなかった
  MSN, AltaVista などの
  サーチエンジンや API を用いる

今後の課題

- Kolmogorov 記述量を用いた改善手法
  1. リンクの重み付けにおいて

     類似度を使った重み付けの式

今後の課題

- Kolmogorov 記述量を用いた改善手法

  2. 距離を求める式において

     圧縮方式の選択

     zip

     bzip2 など他の方式

今後の課題

- Kolmogorov 記述量を用いたの改善手法

  3. タグを利用した既存手法と Kolmogorov 記述量を用いた改善手法を組み合わせる

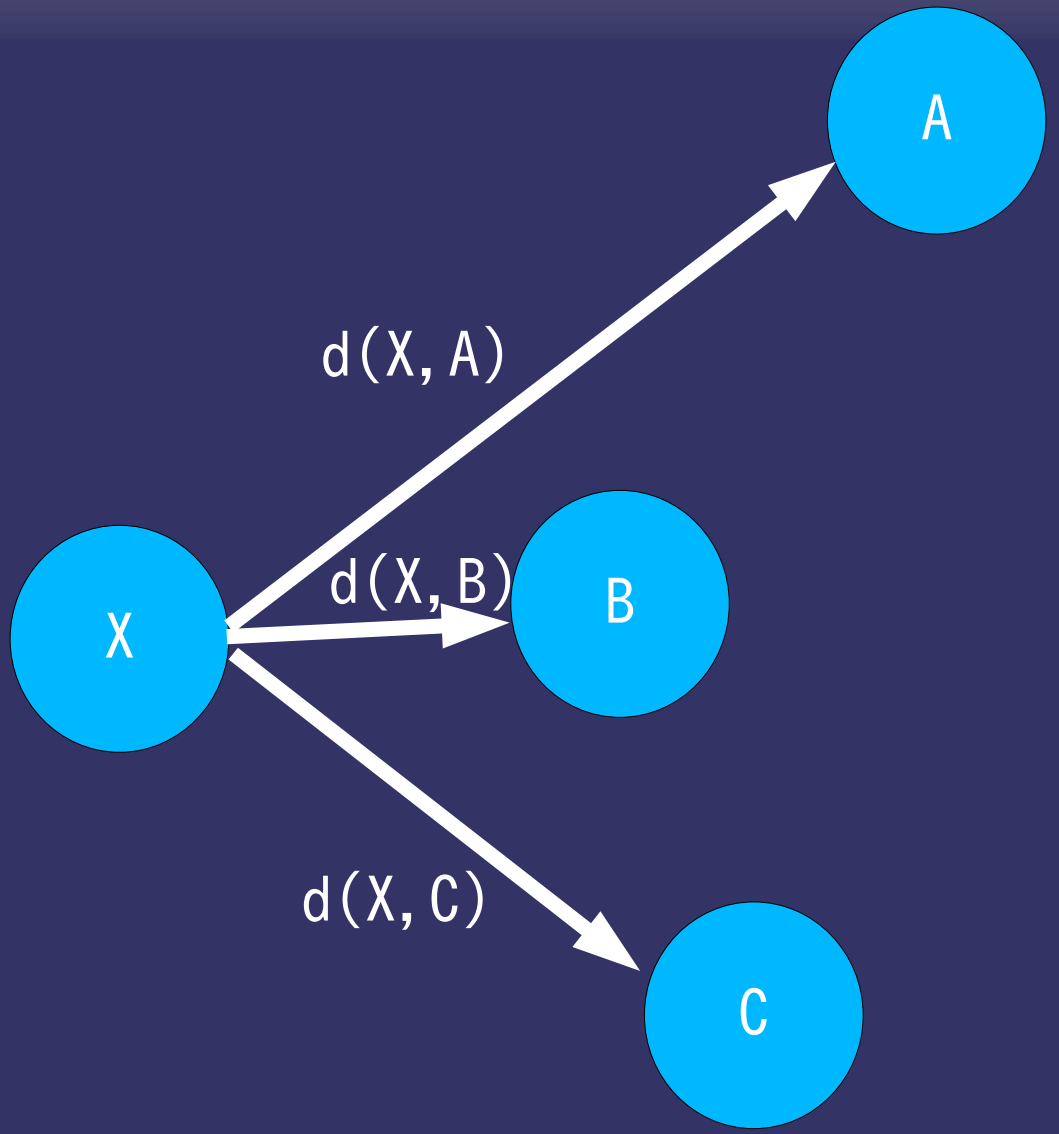
  対象：全てのページ

  一定以上のトピック出現回数のあるページ

終

改善案(Kolmogorov)



d(X,A):X と A の距離

d(X,B):X と B の距離

d(X,C):X と C の距離

*d(i,j):
ページ i と j の
距離

Googleの人気の秘密

支えている物の一部にPageRankという技術が用いられています

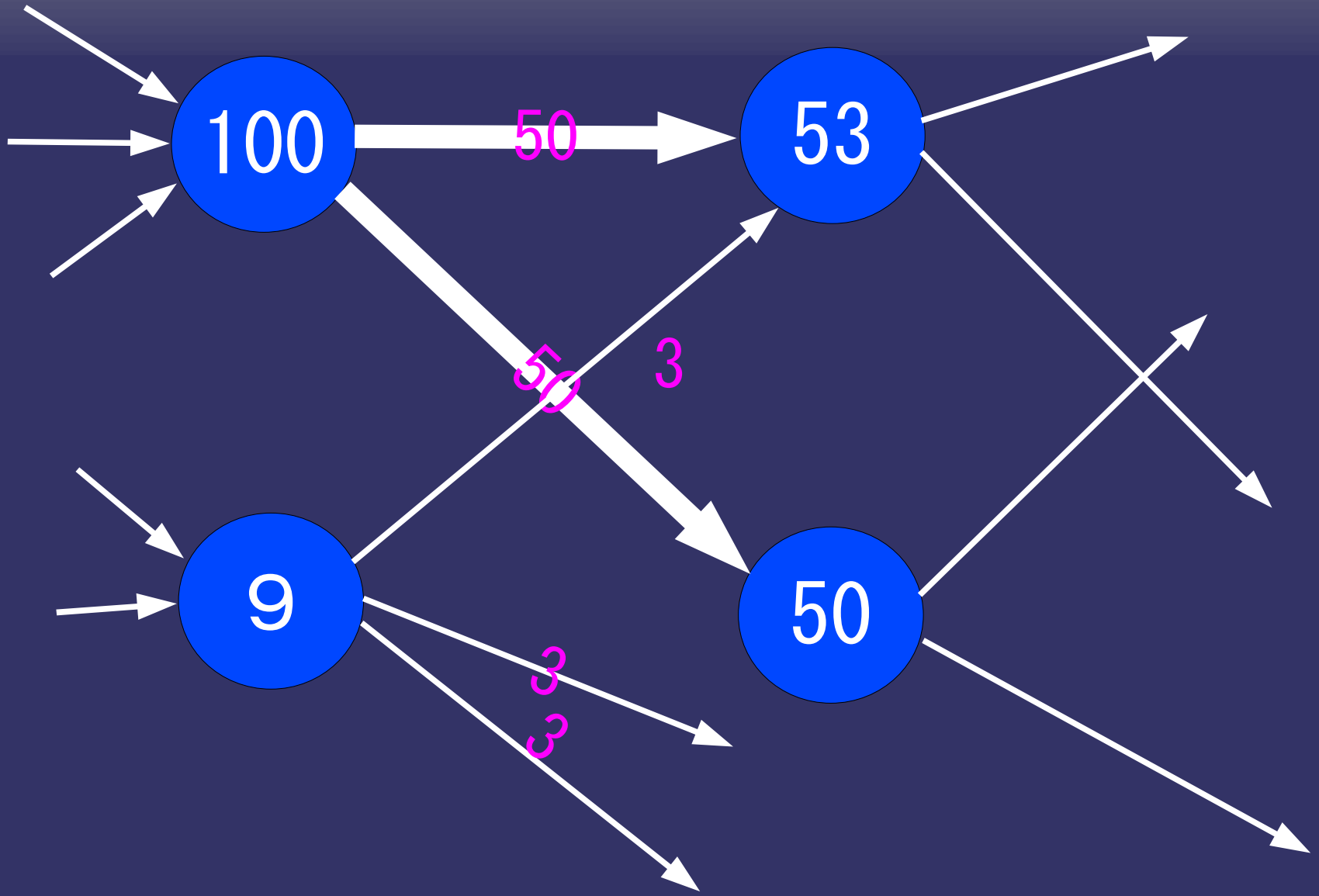
PageRankとは？

「多くの良質なページからされているページは、やはり良質なページである」
という再帰的な関係をもとに
ページの重要度を判定して行くもの

＊ページの重要度：
ページｉからｊへのリンクをｉからｊへの支持投票とみなして、そのページの重要性を判定する

またそのときリンク元の重要度も関係します

PageRank とは？



PageRankとは？

ページの重要度を高めるためには
以下のようなことが考えられる

(1) 被リンク数

(2) 重要度の高いページからのリンク

(3) リンク元のリンク数

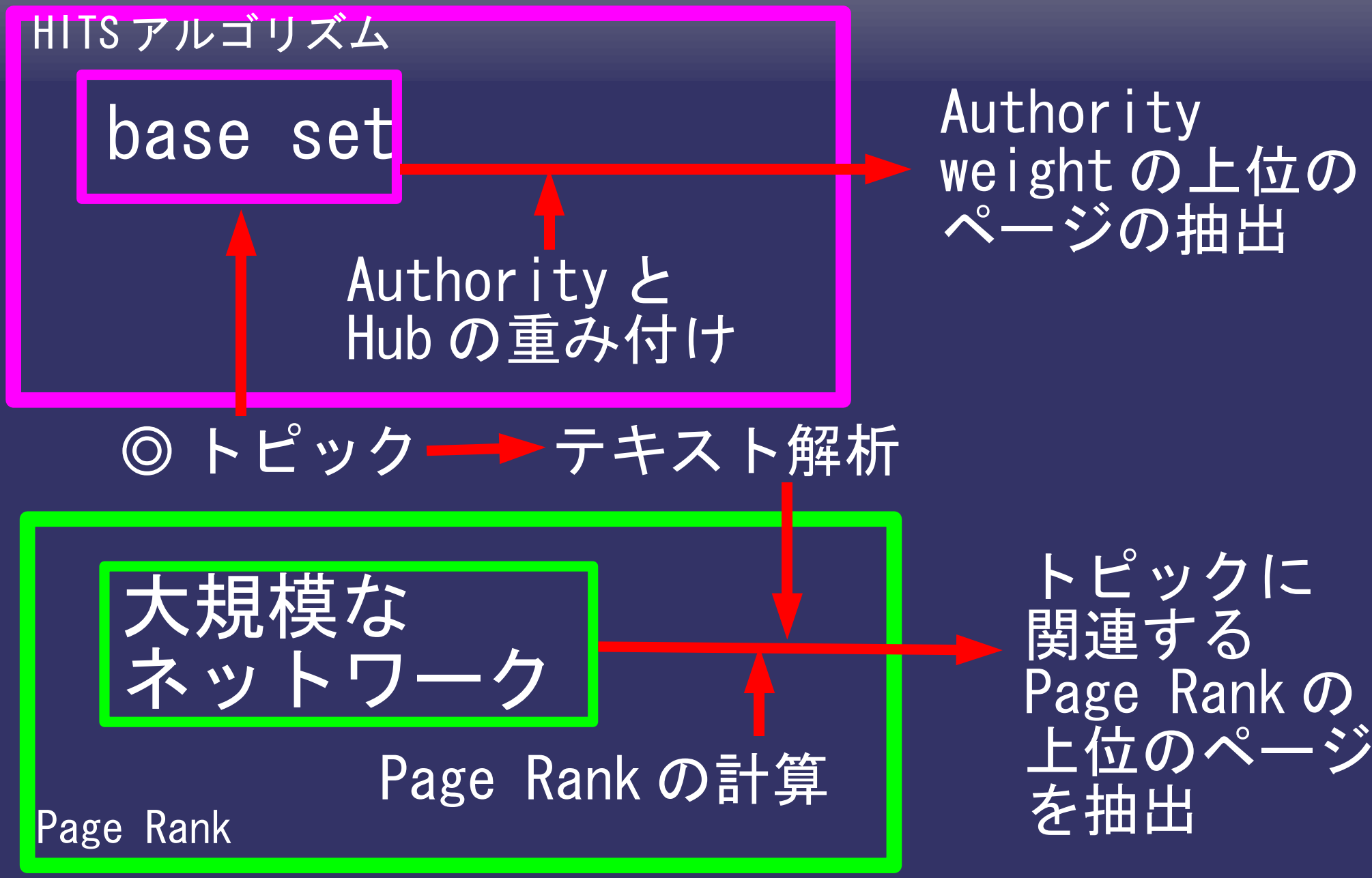
PageRankとは？

ページの重要度をの評価が高いページが
検索順位の順位も上がって来ます

参照ページ
http://www.google.co.jp/intl/ja/why_use.html

http://www.kusastro.kyoto-u.ac.jp/~baba/wais/pagerank.html

PageRank と HITS の違い



反復計算

以上の操作をくりかえし

X ,Y の極限 X^*, Y^*を求める

反復計算

J.Kleinberg らの実験により
極限の値が k=20 から急激に収束し始めることがわかっている

反復回数ｉを２０とする

HITS アルゴリズムの問題点

- base set に検索トピックとは無関係な
  余り適切でないページが含まれ
  そうしたページが密なリンク構造を持って
  いた場合に
  関係ないページの weight を高くしてしまう

- topic drift 問題という

改善案（既存）

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 におけるトピックの頻度

反復計算

以上の操作と正規化をくりかえし

X ,Y の極限 X^*, Y^* を求める

先程の更新ルールを考えてみると

$$X^* = (A^T A)^* X^*$$

$$Y^* = (A A^T)^* Y^*$$

となる
これは固有値問題として考えられる

固有値問題

固有値問題で
固有値を求める計算

$$|\lambda E - A| = 0$$

固有方程式と呼ぶ

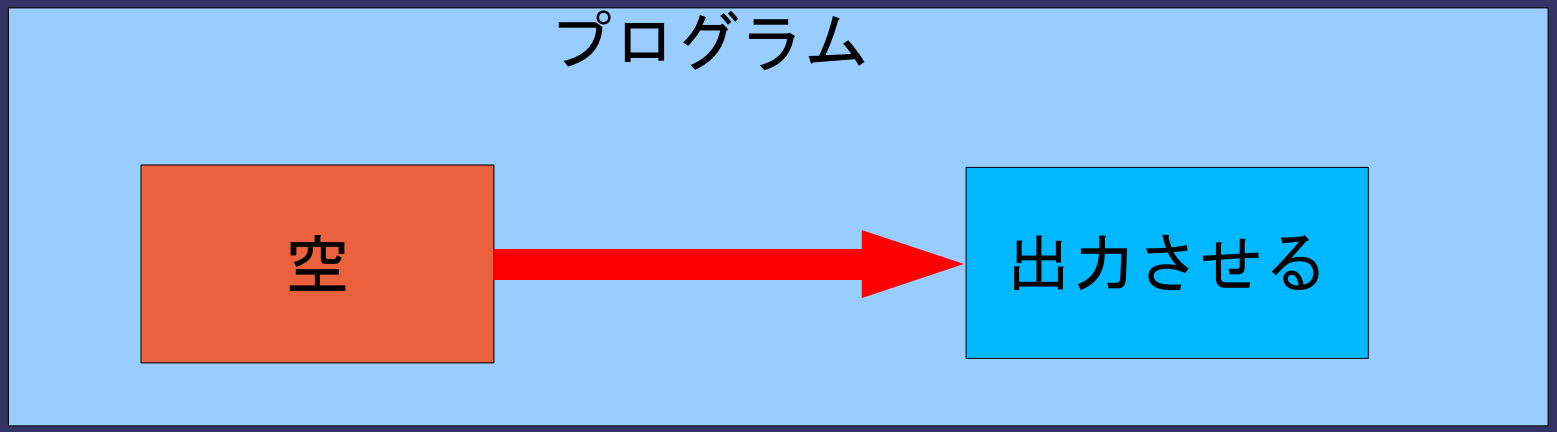
左辺はｎ次多項式となりその解を求めるのは一般的に難しいされています

目次

- 目的
- Kolmogorov 記述量
- HITS アルゴリズムの紹介
- 改善案の紹介
- 今後の課題

Kolmogorov 記述量

前提として
データの情報量をそのデータを作成するプログラムのサイズと定義する



Kolmogorov 記述量

データ x についての Kolmogorov 記述量を

S: プログラム言語
|p|: プログラムのサイズ
S(p): プログラム言語 S で書かれた
プログラム x の出力

$$K_s(x) = min\{ |p| : S(p) = x \}$$

と定義する

＊ x を生成する p が存在しないとき
∞ となる

Kolmogorov 記述量

他の言語 S’ を用いた場合：$K_{s'}(x)$

$$K_s(x) \preceq K_{s'}(x) + c \quad \text{c: 定数}$$

c を無視する

Kolmogorov 記述量を今後

$$K(x)$$

とする

条件付きKolmogorov記述量

補助情報yに対するデータxの
Kolmogorov記述量を

$$K(x \mid y) = min\{ \mid p \mid : U(p, y) = x \}$$

と定義し

条件付きKolmogorov記述量と呼ぶ

＊U：データ記述用に固定した万能言語

情報量

言い替えると
K(x|y) がK(x) に比べて小さいとき
y はx の情報を多く持つことになる

y に含まれるx に関する情報量

$$I(x:y)=K(x)-k(x/y)$$

と定義する

情報量

さらに

$$I(x:y) \lesnsim I(y:x)+c$$

c を無視する

$$I(x:y)=I(y:x)$$

先程の式で書き換えると

$$K(x)+K(y/x)=K(y)+K(x/y)$$

となる

類似度の距離

Kolmogorov 記述量を用いた類似度の距離を

$$d(x,y)=\frac{max\{K(x/y),K(y/x)\}}{max\{K(x),K(y)\}}$$

と定義する

$K(y)AK(x)$のとき

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

類似度の距離

さらに近似的に

$$k(y/x)=K(xy)-K(x)$$

が分かっているので
先程の式を以下のようにします

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

＊ xy ：データ x と y を連結させたファイル

類似度の距離

Kolmogorv 記述量を用いて
類似度の距離を定義しました

実際にはKolmogorov 記述量は計算不可能である

近似式

$$k(x) \notin zip(x)$$

として式を変形させる

＊ zip(x) ： x を圧縮したファイルのサイズ

類似度の距離

よって

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

$$k(x)\notin zip(x)$$

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

類似度の距離

本研究では

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

を用いて
類似度の距離を求めていく