Kleinberg's HITS (Hubs and Authorities) Algorithm

Teodora Dobos

Technical University of Munich

January 2020

Outline

Introduction

Given a query string σ and a search engine ε:

1. How can ε find the most relevant pages to σ?
   - ranking algorithms
2. How can a relevant page p be identified/ranked?
   - boolean models?
   - vector space models?
   - probabilistic models?

Introduction

Given a query string σ and a search engine ε:

1. How can ε find the most relevant pages to σ?
   - ranking algorithms
2. How can a relevant page p be identified/ranked?
   - boolean models?
   - vector space models?
   - probabilistic models?
   - **link analysis - HITS!**

HITS (Hubs and Authorities) Algorithm Overview

- Hyperlink-Induced Topic Search
- Jon Kleinberg, 1998
- IBM CLEVER Project

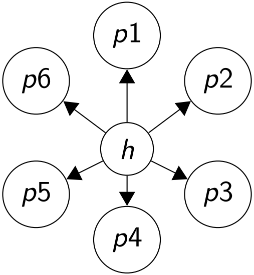
Figure: Web graph [14].

Figure: Jon Kleinberg [9].

Main idea: analyze the link structure of a hyperlinked environment and retrieve the most *authoritative* pages

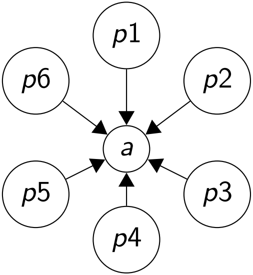
Hubs and Authorities

- Page h contains links to pages $p_1, p_2, p_3, p_4, p_5, p_6$:



h is called **hub.**

- Pages p_1, p_2, p_3, p_4 contain links to page q:



a is called **authority.**

Assign two scores to each page: a hub and an authority score.

HITS Algorithm - Iterative Approach

Given a query σ:

1. Construct a subgraph G_σ of the whole WWW network.
2. Compute iteratively hub and authority scores for each page in G_σ.
3. Pages with the highest authority scores: most relevant to σ.

Subgraph Computation - Vertex Set

Goal: construct an induced subgraph $G_\sigma[V]$ of the web graph at query time.

What is the vertex set V?

Subgraph Computation - Vertex Set

Goal: construct an induced subgraph $G_\sigma[V]$ of the web graph at query time.

What is the vertex set V?

- First idea: $V = Q_\sigma$,
  where $Q_\sigma :=$ set of all pages containing the query string.

Subgraph Computation - Vertex Set

Goal: construct an induced subgraph $G_\sigma[V]$ of the web graph at query time.

What is the vertex set V?

- First idea: $V = Q_\sigma$,
  where $Q_\sigma :=$ set of all pages containing the query string.
- Second idea: $V = S_\sigma$, with S_σ having the properties:
  1. S_σ is small.
  2. S_σ has numerous relevant pages.
  3. S_σ includes most (or many) of the strongest authorities.

Subgraph Computation - Vertex Set

Goal: construct an induced subgraph $G_\sigma[V]$ of the web graph at query time.

What is the vertex set V?

- First idea: $V = Q_\sigma$,
  where $Q_\sigma :=$ set of all pages containing the query string.
- Second idea: $V = S_\sigma$, with S_σ having the properties:
  1. S_σ is small.
  2. S_σ has numerous relevant pages.
  3. S_σ includes most (or many) of the strongest authorities.

The collection of pages S_σ is called the *base set*.

Subgraph Computation

Compute the subgraph G_σ:

Step 1: build the *root set* $R_\sigma \subset Q_\sigma$

- select top $t \approx 200$ highest-ranked pages for σ from a text-based search engine (e.g. AltaVista)
- $G[R_\sigma]$ is sparse - properties 1, 2, 3?

Step 2: apply *BuildSubgraph* procedure (next slide) to obtain G_σ.

Subgraph Computation - Pseudocode

Algorithm BuildSubgraph

Data: R_σ, d
Result: G_σ

Set $S_\sigma = R_\sigma$

forall $p \in R_\sigma$ **do**
    Add all pages in $\Gamma_{out}(p)$ to S_σ
    if $|\ \Gamma_{in}(p)\ | \leq d$ **then**
        Add all pages in $\Gamma_{in}(p)$ to S_σ
    else
        Choose $\Gamma_d(p) \subseteq \Gamma_{in}(p)$ such that $|\ \Gamma_d(p)\ | = d$
        Add $\Gamma_d(p)$ to S_σ

Subgraph Computation

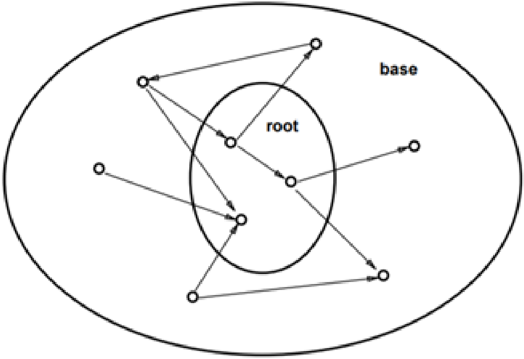


Figure: Root and Base Sets [5].

Hub and Authority Scores Computation

Let $p \in S_\sigma$ be a web page.

- First idea: authority score of p = in-degree of p.

Hub and Authority Scores Computation

Let $p \in S_\sigma$ be a web page.

- First idea: authority score of p = in-degree of p.

  Query example: "java" - pages with largest in-degree:

  - www.gamelan.com
  - java.sun.com
  - other pages advertising Caribbean vacations
  - home page of Amazon Books

  Problem?

Hub and Authority Scores Computation

Let $p \in S_\sigma$ be a web page.

- First idea: authority score of p = in-degree of p.

  Query example: "java" - pages with largest in-degree:

  - www.gamelan.com
  - java.sun.com
  - other pages advertising Caribbean vacations
  - home page of Amazon Books

  Problem?

- Second idea: consider *hub* pages.

Hub and Authority Scores Computation

For each page $p \in S_\sigma$ let:

- $a^{\langle p \rangle}$: authority weight
- $h^{\langle p \rangle}$: hub weight

Invariant: $\sum_{p \in S_\sigma} (a^{\langle p \rangle})^2 = 1$ and $\sum_{p \in S_\sigma} (h^{\langle p \rangle})^2 = 1$.

Hub and Authority Scores Computation

For each page $p \in S_\sigma$ let:

- $a^{\langle p \rangle}$: authority weight
- $h^{\langle p \rangle}$: hub weight

Invariant: $\sum_{p \in S_\sigma} (a^{\langle p \rangle})^2 = 1$ and $\sum_{p \in S_\sigma} (h^{\langle p \rangle})^2 = 1$.

Given weights $\{a^{\langle p \rangle}\}$ and $\{h^{\langle p \rangle}\}$, define operations:

$$\mathcal{I}: \; a^{\langle p \rangle} = \sum_{q:(q,p) \in E} h^{\langle q \rangle}$$

$$\mathcal{O}: \; h^{\langle p \rangle} = \sum_{q:(p,q) \in E} a^{\langle q \rangle}.$$

Hub and Authority Scores Computation

For each page $p \in S_\sigma$ let:

- $a^{\langle p \rangle}$: authority weight
- $h^{\langle p \rangle}$: hub weight

Invariant: $\sum_{p \in S_\sigma} (a^{\langle p \rangle})^2 = 1$ and $\sum_{p \in S_\sigma} (h^{\langle p \rangle})^2 = 1$.

Given weights $\{a^{\langle p \rangle}\}$ and $\{h^{\langle p \rangle}\}$, define operations:

$$\mathcal{I}: \ a^{\langle p \rangle} = \sum_{q:(q,p) \in E} h^{\langle q \rangle}$$

$$\mathcal{O}: \ h^{\langle p \rangle} = \sum_{q:(p,q) \in E} a^{\langle q \rangle}.$$

Mutually reinforcing relationship between hubs and authorities.

Authority and Hub Scores Computation

$a[p] := h[q1] + h[q2] + h[q3]$ $\qquad$ $h[p] := a[q1] + a[q2]$

Figure: Example.

HITS Algorithm - Pseudocode

Algorithm HITS

Data: G_σ, k, c
Result: a_k, h_k, best_authorities, best_hubs
Set $a_0 = z$
Set $h_0 = z$
for $i \leftarrow 1$ to k **do**
  Apply $\mathcal{I}$ to (a_{i-1}, h_{i-1}) and obtain new a-weights a'_i
  Apply $\mathcal{O}$ to (a'_i, h_{i-1}) and obtain new h-weights h'_i
  Set a_i = Normalize (a'_i)
  Set h_i = Normalize (h'_i)
for $j \leftarrow 1$ to c **do**
  best_authorities[j] = getMaxAndRemoveIt(a_k)
  best_hubs[j] = getMaxAndRemoveIt(h_k)

* $z = (1, 1, ..., 1)$

Hub and Authority Scores - Matrix Vector Products

Recall: $A \in \mathbb{N}^{n \times n}$ adjacency matrix of G_σ, $|S_\sigma| = n$, a authority vector, h hub vector

Then:

$$\mathcal{I} : a \leftarrow A^T h \tag{1}$$

$$\mathcal{O} : h \leftarrow Aa. \tag{2}$$

Set initial scores $a_0 = h_0 = (1, 1, ..., 1) = \mathbf{1}$ and apply (1) and (2):

$$a_k = A^T AA^T AA^T A ... A^T AA^T \mathbf{1} = (A^T A)^{k-1} A^T \mathbf{1},$$

$$h_k = AA^T AA^T AA^T ... AA^T \mathbf{1} = (AA^T)^k \mathbf{1}.$$

Linear Algebra Notions

Let $M \in \mathbb{R}^{n \times n}$ be a **symmetric** matrix, λ an eigenvalue of M and ω a vector such that:

$$M\omega = \lambda\omega.$$

Then:

- $E = \{\omega : M\omega = \lambda\omega\}$ is the eigenspace of M associated to λ.
- $\mu_M(\lambda) = \gamma_M(\lambda)$.
- $dim(E) = \mu_M(\lambda) = \gamma_M(\lambda)$.
- M has at most n distinct real eigenvalues $\lambda_1(M)$, $\lambda_2(M)$,..., $\lambda_n(M)$ (indexed in order of decreasing absolute value) and $\sum_{i=1}^{n} \mu_M(\lambda_i) = n$.

Perron-Frobenius Theorem

1. The largest eigenvalue λ_1 of M (spectral radius $\rho(M)$) is positive and has multiplicity 1.

2. Each other eigenvalue of M is in modulus strictly less than λ_1:

$$|\,\lambda_1(M)\,| > |\,\lambda_2(M)\,| \geq \ \ldots \ \geq |\,\lambda_n(M)\,|.$$

3. The largest eigenvalue λ_1 has a corresponding eigenvector $\omega_1(M)$ with **all entries positive**. $\omega_1(M)$ is the *principal eigenvector* of M.

Convergence

Assumption: The sequences $a_1, a_2, \ldots, a_k$ and $h_1, h_2, \ldots, h_k$ converge to limits a^* and h^* respectively.

Convergence

Assumption: The sequences $a_1, a_2, \dots, a_k$ and $h_1, h_2, \dots, h_k$ converge to limits a^* and h^* respectively.

Proof:

- Matrices A^TA and AA^T are symmetric and have real eigenvalues.
- a_k is the unit vector in the direction $(A^TA)^{k-1}A^T\mathbf{1}$.
- h_k is the unit vector in the direction $(AA^T)^k\mathbf{1}$.
- Lemma: If M is a symmetric $n \times n$ matrix and v is a vector not orthogonal to the principal eigenvector $\omega_1(M)$, then the unit vector in the direction of $M^k v$ converges to $\omega_1(M)$ as k increases without bound $(*)$.
- Set $M = AA^T$ and $v = \mathbf{1}$.
- It follows that the sequence $h_1, h_2, \dots, h_k$ converges to $\omega_1(AA^T)$.

Convergence

- $\lambda_1(A^T A) \neq 0$.
- $A^T \mathbf{1}$ is not orthogonal to $\omega_1(A^T A)$.
- Set $M = A^T A$ and $v = A^T \mathbf{1}$ $(*)$.
- It follows that the sequence $a_1, a_2, \ldots, a_k$ converges to $\omega_1(A^T A)$.

The hub and authority scores converge to the principal eigenvectors of AA^T and $A^T A$ respectively.

Principal Eigenvectors Computation - Power Iteration

Algorithm Power Iteration

Data: $M, x^{(0)}, k$
Result: $x^{(k)}, \lambda^{(k)}$
for $i \leftarrow 1$ to k **do**

$$w^{(i)} = Mx^{(i)}$$
$$\lambda^{(i)} = (x^{(i)})^T w^{(i)}$$
$$x^{(i+1)} = \frac{w^{(i)}}{\|w^{(i)}\|}$$

Complexity: $O(n^2)$

Evaluation

(java) Authorities
.328 http://www.gamelan.com/
(Gamelan)
.251 http://java.sun.com/
(JavaSoft Home Page)
.190 http://www.digitalfocus.com/digitalfocus/faq/howdoi.html
(The Java Developer: How Do I...)
.190 http://lightyear.ncsa.uiuc.edu/srp/java/javabooks.html
(The Java Book Pages)
.183 http://sunsite.unc.edu/javafaq/javafaq.html
(comp.lang.java FAQ)

Strengths and Weaknesses

Strengths:

1. space efficiency
2. HITS is sensitive to user query

Weaknesses:

1. high computational cost at query time
2. tightly-knit community effect (TKC) and topic drift problem
3. operations $\mathcal{I}$ and $\mathcal{O}$ must be executed on the fly at query time
4. small robustness to spam
5. HITS cannot identify irrelevant authorities
6. HITS cannot identify irrelevant hubs

Improvements - Bloom Filters-Based Approach

Problem: high computational cost at query time required for computing G_σ

Improvements - Bloom Filters-Based Approach

Problem: high computational cost at query time required for computing G_σ

Solution: move the most expensive part of the computation offline

Improvements - Bloom Filters-Based Approach

Problem: high computational cost at query time required for computing G_σ

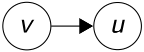
Solution: move the most expensive part of the computation offline

- *index-construction time*: create a database: map web page URLs to summaries of their neighborhoods (consistent sampling - deterministic)
- *query time*:
  - look up each page of the root set in the summary database
  - approximate the neighborhood graph
  - compute hub and authority scores

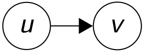
Summary of the Neighborhood Graph of A Web Page

Given two web pages u and v:

- v is an *ancestor* of u:



- v is a *descendant* of u:



Summary of the neighborhood graph of u = summary of the ancestors (Bloom filter) & summary of the descendants (Bloom filter) of u

Bloom Filter

- space-efficient probabilistic data structure
- test of the membership of an element in a given collection
- array F of m bits
- k hash functions $h_1.h_2, ..., h_k \implies$ set of values: $[1, m]$
- false positive matchings are possible ☹, but false negatives are not ☺
- add e to F: $\forall i \in [k] : F[h_i(e)] = 1$ (impossible remove)

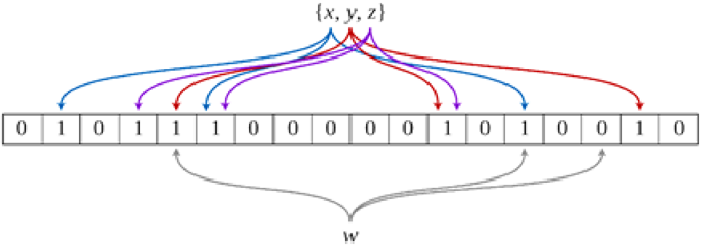


Figure: Bloom Filter [15].

Summary Computation

- $BF[X]$: Bloom filter representing set X
- $C_n[X]$:= consistent unbiased sample of elements, $C_n[X] = X$ if $|X| < n$
- $I_n(u) = C_n[v \in V : (v, u) \in E]$: consistent sample of (at most) n ancestors
- $O_n(u) = C_n[v \in V : (u, v) \in E]$ be a consistent sample of n descendants

Summary Computation

- $BF[X]$: Bloom filter representing set X
- $C_n[X]$:= consistent unbiased sample of elements, $C_n[X] = X$ if $|X| < n$
- $I_n(u) = C_n[v \in V : (v,u) \in E]$: consistent sample of (at most) n ancestors
- $O_n(u) = C_n[v \in V : (u,v) \in E]$ be a consistent sample of n descendants

Idea: compute summary $(BF[I_a(u)], I_b(u), BF[O_c(u)], O_d(u))$ for each page u in the web graph (at **index-construction time**!).

Neighborhood Graph Computation

- (approximate) neighborhood graph $N = (C, E)$
- construct *cover set* : $C = R \cup \bigcup_{u \in R} I_b(u) \cup \bigcup_{u \in R} O_d(u)$ (**query time!**)
- compute edge set E: given $u \in R$ and $v \in C$
  - if $BF[I_a(u)]$ contains v: add (v, u) to E
  - if $BF[O_c(u)]$ contains v: add (u, v) to E

Use N to compute hub and authority scores.

Is N identical to G_σ?

Is N identical to G_σ?

No!

- N is smaller than G_σ.
- N contains edges pointing from $C \cap I_c(u)$ to $u \in R$ and from $u \in R$ to $C \cap O_c(u)$.
- N was constructed using Bloom filters.

Is N identical to G_σ?

No!

- N is smaller than G_σ.
- N contains edges pointing from $C \cap I_c(u)$ to $u \in R$ and from $u \in R$ to $C \cap O_c(u)$.
- N was constructed using Bloom filters.

So:

- sampling process $\Longrightarrow \begin{cases} \text{exclusion of specific edges} \\ \text{inclusion of "phantom" edges} \end{cases}$

But...

Is N identical to G_σ?

No!

- N is smaller than G_σ.
- N contains edges pointing from $C \cap I_c(u)$ to $u \in R$ and from $u \in R$ to $C \cap O_c(u)$.
- N was constructed using Bloom filters.

So:

- sampling process $\Longrightarrow \begin{cases} \text{exclusion of specific edges} \\ \text{inclusion of "phantom" edges} \end{cases}$

But...

Consistent sampling preserves co-citation!

Improved HITS (I-HITS)

Problem: topic drift

Improved HITS (I-HITS)

Problem: topic drift

Solution: compute the *similarity* and *popularity* of pages in the base set

Improved HITS (I-HITS)

Problem: topic drift

Solution: compute the *similarity* and *popularity* of pages in the base set

Define $S_p :=$ similarity between p and σ.

- cosine similarity between p and σ
- if $i \rightarrow j$: similarity between the anchor text and σ

Improved HITS (I-HITS)

Problem: topic drift

Solution: compute the *similarity* and *popularity* of pages in the base set

Define $S_p :=$ similarity between p and σ.

- cosine similarity between p and σ
- if $i \to j$: similarity between the anchor text and σ

Recall: A is the adjacency matrix of the subgraph.

$$A_{i,j} = \begin{cases} (1+S_i)\cdot(1+S_j) & \text{if } i \to j \\ 0 & \text{otherwise.} \end{cases}$$

Popularity of A Page

HITS: quantitatively measure
I-HITS: qualitatively and quantitatively measures

Popularity of A Page

HITS: quantitatively measure
I-HITS: qualitatively and quantitatively measures

Page p points to k pages $q_1, q_2, ..., q_k$.

- p assigns popularity-scores $W_{(p,q_i)}$ to each $q_i, i \in [k]$ such that $\sum_{i \in [k]} W_{(p,q_i)} = 1$.
- $W_{(p,q_i)}$ is calculated based on the "hubness" (W_{out}) or on the "authoritiness" (W_{in}) of q_i:

$$W^{out}_{(j,i)} = \frac{O_i}{\sum_{p \in R(j)} O_p}$$

$$W^{in}_{(j,i)} = \frac{I_i}{\sum_{p \in R(j)} I_p}$$

$I_i = deg_{in}(i), O_i = deg_{out}(i), R(j) :=$ set of pages to which j points

Popularity - Example

- Popularity as a hub:
  - $W^{out}_{(A,C)} = O_C/(O_C + O_D) = 2/(2+3) = 2/5$
  - $W^{out}_{(A,D)} = O_D/(O_D + O_C) = 3/(3+2) = 3/5$
- Popularity as an authority:
  - $W^{in}_{(A,C)} = I_C/(I_C + I_D) = 2/(2+1) = 2/3$
  - $W^{in}_{(A,D)} = I_D/(I_D + I_C) = 1/(1+2) = 1/3$

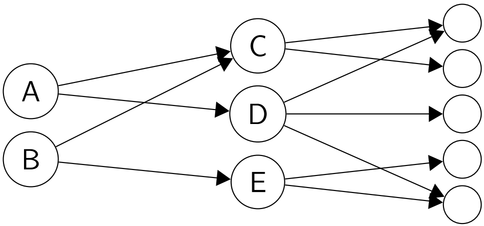


Figure: Example of a linked structure of the web [8].

Update Operations

Update authority score:

$$\mathcal{I}' : a^{\langle i \rangle} = \sum_{j \in B(i)} h_j \cdot (1 + s_i) \cdot (1 + s_{ji}) \cdot \frac{I(i)}{\sum_{p \in F(j)} I(p)}.$$

Update hub score:

$$\mathcal{O}' : h^{\langle i \rangle} = \sum_{j \in F(i)} a_j \cdot (1 + s_i) \cdot (1 + s_{ij}) \cdot \frac{O(i)}{\sum_{p \in B(j)} O(p)}.$$

$B(i)$:= set of pages that contain links to i
$F(i)$:= set of pages to which i contains links

I-HITS - Pseudocode

Algorithm I-HITS

Data: G_σ

Result: a_i, h_i

for $p \in S_\sigma$ **do**

  Set $a_0^{\langle p \rangle} = 1$

  Set $h_0^{\langle p \rangle} = 1$

i = 0

while *a_i and h_i do not converge* **do**

  Set i = i + 1

  forall $p \in S_\sigma$ **do**

    Apply $\mathcal{I}'$ and obtain new authority value $a_i^{\langle p \rangle}$

    Apply $\mathcal{O}'$ and obtain new hub value $h_i^{\langle p \rangle}$

Evaluation

A page can be:

1. **Highly relevant** (HR): high authority score, very important information.
2. *Relevant* (R): relevant, but not important information.
3. Not-relevant (NR): no keywords of the query, no relevant information.

Table: Experimental Data

Query	Nodes	Hubs	Authorities	Links
alcohol	1964	1441	1213	11083
"搜索引擎" (search engine)	2884	2142	1744	37941

Evaluation

Results computed by **HITS** - Query "search engine":

(1) **https://www.google.com/?hl=zh-CN**
(2) *http://www.gseeker.com/*
(3) http://www.wangtam.com/50226711/c_wav
(4) http://www.yuleguan.com/
(5) https://www.chinaventurenews.com/
(6) http://www.tjacobi.com/
(7) http://www.money-courier.com/ (this website no longer exists)
(8) http://www.geekervision.com/
(9) http://www.in-women.com/
(10) https://www.tracingadget.com/ (this website no longer exists)

Evaluation

Results computed by **I-HITS** - Query "search engine":

(1) *http://www.xpue.net* (this website no longer exists)
(2) *http://www.tooooold.com/* (this website no longer exists)
(3) *http://www.bbssearch.cn/* (this website no longer exists)
(4) **https://www.google.com/?hl=zh-CN**
(5) http://www.1hd.cn/ (this website no longer exists)
(6) **http://www.baidu.com/**
(7) *http://bizsite.sina.com.cn/* (this website no longer exists)
(8) http://it.sohu.com/7/0903/35/column213613 (this website no longer exists)
(9) **http://www.youdao.com/?keyfrom=so163redir**
(10) https://www.qq.com/?froma

A Stochastic Approach for Link-Structure Analysis (SALSA)

- addresses the TKC effect and the topic drift problem
- based on stochastic properties of random walks
- two Markov chains: a chain of hubs and a chain of authorities
- identical subgraph computation step

SALSA

- $s \implies s_a$ and s_h
- bipartite undirected graph $\hat{G} = (V_h, V_a, E)$
  - $V_h = \{s_h \mid s_h \in S$ and out-degree$(s_h) > 0\}$
  - $V_a = \{s_a \mid s_a \in S$ and in-degree$(s_a) > 0\}$
  - $E = \{\{s_h, r_a\} \mid s_h \in V_h,\ r_a \in V_a$ and $s_h \to r_a$ in $S\}$

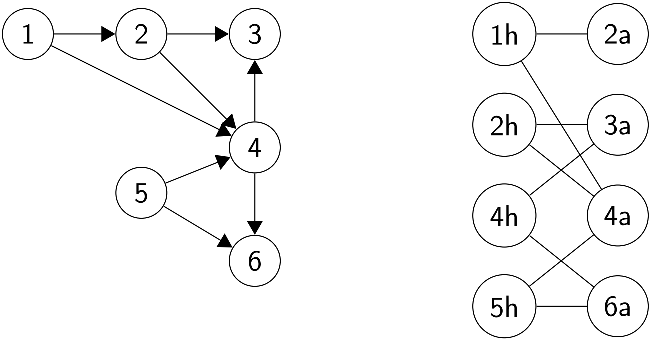


Figure: Transforming the graph on the left side into a bipartite graph [6].

Two Markov Chains

- $(X_n)_{n=0}^{\infty}$ with state space V_a and $(Y_n)_{n=0}^{\infty}$ with state space V_h
- random walks, but not in the "normal" sense: state transitions are generated by traversing two links in a row, one link forward and one link backwards (example?)
- start off from different sides of $\hat{G}$

Transition Matrices

- authority chain - stochastic matrix $\hat{A}$ with:

$$\hat{a}_{i,j} = \sum_{\{k|(k_h,i_a),(k_h,j_a)\in\hat{G})\}} \frac{1}{deg(i_a)} \cdot \frac{1}{deg(k_h)}.$$

- hub chain - stochastic matrix $\hat{H}$ with:

$$\hat{h}_{i,j} = \sum_{\{k|(i_h,k_a),(j_h,k_a)\in\hat{G})\}} \frac{1}{deg(i_h)} \cdot \frac{1}{deg(k_a)}.$$

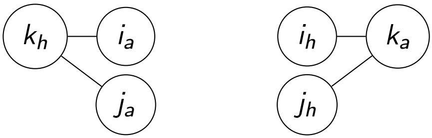


Figure: Visual representation of positive probabilities $\hat{a}_{i,j}$ (left) and $\hat{h}_{i,j}$ (right).

Best Hubs and Best Authorities

Assumption: The principal eigenvectors of $\hat{H}$ and $\hat{A}$ contain the scores corresponding to the best hubs and authorities respectively.

Best Hubs and Best Authorities

Assumption: The principal eigenvectors of $\hat{H}$ and $\hat{A}$ contain the scores corresponding to the best hubs and authorities respectively.

Proof:

- both Markov chains are irreducible and aperiodic $\implies$ ergodic MCs
- Ergodic Theorem: the principal eigenvector of an ergodic Markov chain is its stationary distribution.
- π := the stationary distribution of the chain of authorities
- a_n := the distribution of this chain for the n-th step of the RW
- then:

$$\lim_{n\to\infty} a_n = \pi.$$

Best Hubs and Best Authorities

Recall: A is the adjacency matrix associated to the subgraph.

- A_r: the matrix which results by dividing each nonzero entry by the sum of the entries in its row
- A_c: the matrix obtained by dividing each nonzero entry by the sum of the entries in its column

Best Hubs and Best Authorities

Recall: A is the adjacency matrix associated to the subgraph.

- A_r: the matrix which results by dividing each nonzero entry by the sum of the entries in its row
- A_c: the matrix obtained by dividing each nonzero entry by the sum of the entries in its column

Then:

- $\hat{H}$: nonzero rows and columns of $A_r A_c^T$
- $\hat{A}$: nonzero rows and columns of $A_c^T A_r$

Evaluation

Authorities computed by **HITS** - Query "movies":

.1673 http://go.msn.com/npl/msnt.asp
(MSN.COM)
.1672 http://go.msn.com/bql/whitepages/asp
(White Pages - msn.com)
.1672 http://go.msn.com.nsl/webevents.asp
(Web Events)
.1672 http://go.msn.com/bql/scoreboards.asp
(MSN Sports scores)

Evaluation

Hubs computed by **HITS** - Query "movies":

.1692 http://denver.sidewalk.com/movies
(movies: denver.sidealk)
.1619 http://boston.sidewalk.com/movies
(movies: boston.sidewalk)
.1688 http://twincities.sidewalk.com/movies
(movies: twincities.sidewalk)
.1686 http://newyork.sidewalk.com/movies
(movies: newyork.sidewalk)

Evaluation

Authorities computed by **SALSA** - Query "movies":

.2533 http://us.imdb.com/
(The Internet Movie Database)
.2233 http://www.mrshowbiz.com/
(Mr Showbiz)
.2200 http://www.disney.com/
(Disney.com-The Web Site for Families)
.2134 http://www.hollywood.com/
(Hollywood Online:...all about movies)
.2000 http://www.imdb.com/
(The Internet Movie Database)
.1967 http://www.paramount.com/
(Welcome to Paramount Pictures)
.1800 http://www.mca.com/
(Universal Studios)

Conclusion

- link analysis algorithm
- hub and an authority scores
- repeated improvement
- focused subgraph constructed at query time
- final scores: principal eigenvectors of hub and authority matrices
- important weaknesses:
  - computational effort at query time
  - TKC effect
  - topic drift problem

References I

Giorgio Ausiello and Rossella Petreschi Eds. *The Power of Algorithms - Inspiration and Examples in Everyday Life*. Springer-Verlag, 2013.

Dr. Klaus Berberich and Dr. Pauli Miettinen. *HITS*. URL: http://resources.mpi-inf.mpg.de/departments/d5/teaching/ws13_14/irdm/slides/irdm-4-3.pdf. (date accessed: 26.11.2019).

Ulrik Brandes and Thomas Erlebach (Eds.). *Network Analysis - Methodological Foundations*. Springer-Verlag, 2005.

Sreenivas Gollapudi, Marc Najork, and Rina Panigrahy. *Using Bloom Filters to Speed Up HITS-like Ranking Algorithms*. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/waw2007.pdf. (date accessed: 30.10.2019).

References II

Jon M. Kleinberg. *Authoritative Sources in a Hyperlinked Environment*. URL: http://www.cs.cornell.edu/home/kleinber/auth.pdf. (date accessed: 23.10.2019).

R. Lempel and S. Moran. *The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect*. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.5859&rep=rep1&type=pdf. (date accessed: 28.11.2019).

Jure Leskovec. *Link Analysis: PageRank and Similar Idea*. URL: http://snap.stanford.edu/class/cs246-2012/slides/10-hits.pdf. (date accessed: 29.10.2019).

References III

Xinyue Liu, Hongfei Lin, and Cong Zhang. *An Improved HITS Algorithm Based on Pagequery Similarity and Page Popularity*. URL: https://pdfs.semanticscholar.org/e7e1/82659614da4f92daca8d8455fd11350f198a.pdf. (date accessed: 31.10.2019).

Net Worker. URL: https://www.smithsonianmag.com/innovation/net-worker-167228451/. (date accessed: 10.12.2019).

Saeko Nomura et al. *Analysis and Improvement of HITS Algorithm for Detecting Web Communities*. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.5271&rep=rep1&type=pdf. (date accessed: 31.10.2019).

References IV

Punit Patel and Kanu Patel. *A Review of PageRank and HITS Algorithms*. URL: http://ijarest.com/papers/finished_papers/A%20Review%20of%20PageRank%20and%20HITS%20Algorith.pdf. (date accessed: 24.10.2019).

Enoch Peserico and Luca Pretto. *Score and Rank Convergence of HITS*. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.415.843&rep=rep1&type=pdf. (date accessed: 09.11.2019).

Raluca Remus. *Lecture #4: HITS Algorithm - Hubs and Authorities on the Internet*. URL: http://pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture4/lecture4.html. (date accessed: 24.10.2019).

References V

Taking Network Analysis Beyond Email. URL: https://www.worklytics.co/going-beyond-email-in-organizational-network-analysis/. (date accessed: 10.12.2019).

Wikipedia. *Bloom filter*. URL: https://en.wikipedia.org/wiki/Bloom_filter. (date accessed: 30.10.2019).

Wikipedia. *CLEVER project*. URL: https://en.wikipedia.org/wiki/CLEVER_project. (date accessed: 26.11.2019).

Wikipedia. *HITS algorithm*. URL: https://en.wikipedia.org/wiki/HITS_algorithm. (date accessed: 29.10.2019).

References VI

Wikipedia. *Perron–Frobenius theorem*. URL: https://en.wikipedia.org/wiki/Perron-Frobenius_theorem. (date accessed: 24.10.2019).

Wikipedia. *Power Iteration*. URL: https://en.wikipedia.org/wiki/Power_iteration. (date accessed: 30.10.2019).

Wikipedia. *SALSA algorithm*. URL: https://en.wikipedia.org/wiki/SALSA_algorithm. (date accessed: 28.11.2019).

Yanchun Zhang, Jeffrey Xu Yu, and Jingyu Hou. *Web Communities: Analysis and Construction*. Springer-Verlag, 2006.