



WIKIPEDIA
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

[What links here](#)

[Related changes](#)

[Upload file](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Wikidata item](#)

[Cite this page](#)

Print/export

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

Languages

[العربية](#)

[Català](#)

[Deutsch](#)

[Español](#)

[Euskara](#)

[Français](#)

[Magyar](#)

[Polski](#)

[Русский](#)

[Edit links](#)

Article [Talk](#)

[Read](#)

[Edit](#)

[View history](#)



HITS algorithm

From Wikipedia, the free encyclopedia

(Redirected from [Hyperlink-Induced Topic Search](#))

Hyperlink-Induced Topic Search (**HITS**; also known as **hubs and authorities**) is a [link analysis algorithm](#) that rates Web pages, developed by [Jon Kleinberg](#). The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that it held, but were used as compilations of a broad catalog of information that led users directly to other authoritative pages. In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs.^[1]

The scheme therefore assigns two scores for each page: its authority, which estimates the value of the content of the page, and its hub value, which estimates the value of its links to other pages.

Contents [hide]

1 History

1.1 In journals

1.2 On the Web

2 Algorithm

3 In detail

3.1 Authority Update Rule

3.2 Hub Update Rule

3.3 Normalization

4 Pseudocode

5 Non-converging pseudocode

6 References

7 External links

History [\[edit\]](#)

In journals [\[edit\]](#)

Formerly, many methods were used for ranking the importance of scientific journals. One such method was Garfield's [impact factor](#). However, many journals such as Science and Nature are filled with numerous citations, making these magazines have very high impact factors. Thus, when comparing two more obscure journals which have received roughly the same number of citations but one of these journals has received many citations from Science and Nature, this journal needs be ranked higher. In other words, it is better to receive citations from an important journal than from an unimportant one.^[2]

On the Web [\[edit\]](#)

This phenomenon also occurs in the [Internet](#). Counts the number of links to a page can give us a general estimate of its prominence on the Web, but a page with very few incoming links may also be prominent, if two of these links come from the home pages of [Yahoo!](#) or [Google](#) or [MSN](#). Thus, because these sites are of very high importance but are also Search Engines, there can be very irrelevant results. The [Twitter](#) Social Network uses a HITS style algorithm to suggest user accounts to follow.^[3]

Algorithm [\[edit\]](#)

In the HITS algorithm, the first step is to retrieve the most relevant pages to the search query. This set is called the *root set* and can be obtained by taking the top n pages returned by a text-based search algorithm. A *base set* is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it. The web pages in the base set and all hyperlinks among those pages form a focused subgraph. The HITS computation is performed only on this *focused subgraph*. According to Kleinberg the reason for constructing a base set is to ensure that most (or many) of the strongest authorities are included.

Authority and hub values are defined in terms of one another in a [mutual recursion](#). An authority value is

computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Some implementations also consider the relevance of the linked pages.

The algorithm performs a series of iterations, each consisting of two basic steps:

- **Authority Update:** Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.
- **Hub Update:** Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

- Start with each node having a hub score and authority score of 1.
- Run the Authority Update Rule
- Run the Hub Update Rule
- Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
- Repeat from the second step as necessary.

HITS, like [Page](#) and [Brin's PageRank](#), is an [iterative algorithm](#) based on the [linkage of the documents on the web](#). However it does have some major differences:

- It is query dependent, that is, the (Hubs and Authority) scores resulting from the link analysis are influenced by the search terms;
- As a corollary, it is executed at query time, not at indexing time, with the associated hit on performance that accompanies query-time processing.
- It is not commonly used by search engines. (Though a similar algorithm was said to be used by [Teoma](#), which was acquired by [Ask Jeeves/Ask.com](#).)
- It computes two scores per document, hub and authority, as opposed to a single score;
- It is processed on a small subset of 'relevant' documents (a 'focused subgraph' or base set), not all documents as was the case with PageRank.

In detail [\[edit\]](#)

To begin the ranking, $\forall p, \text{auth}(p) = 1$ and $\text{hub}(p) = 1$. We consider two types of updates: Authority Update Rule and Hub Update Rule. In order to calculate the hub/authority scores of each node, repeated iterations of the Authority Update Rule and the Hub Update Rule are applied. A k-step application of the Hub-Authority algorithm entails applying for k times first the Authority Update Rule and then the Hub Update Rule.

Authority Update Rule [\[edit\]](#)

$\forall p$, we update $\text{auth}(p)$ to be the summation:

$$\text{auth}(p) = \sum_{i=1}^n \text{hub}(i)$$

where n is the total number of pages connected to p and i is a page connected to p. That is, the Authority score of a page is the sum of all the Hub scores of pages that point to it.

Hub Update Rule [\[edit\]](#)

$\forall p$, we update $\text{hub}(p)$ to be the summation:

$$\text{hub}(p) = \sum_{i=1}^n \text{auth}(i)$$

where n is the total number of pages p connects to and i is a page which p connects to. Thus a page's Hub score is the sum of the Authority scores of all its linking pages

Normalization [\[edit\]](#)

The final hub-authority scores of nodes are determined after infinite repetitions of the algorithm. As directly and iteratively applying the Hub Update Rule and Authority Update Rule leads to diverging values, it is necessary to [normalize](#) the matrix after every iteration. Thus the values obtained from this process will eventually converge.^[4]

Pseudocode [\[edit\]](#)

```
1  $G :=$  set of pages
2 for each page  $p$  in  $G$  do
3    $p.auth = 1$  //  $p.auth$  is the authority score of the page  $p$ 
4    $p.hub = 1$  //  $p.hub$  is the hub score of the page  $p$ 
5 function HubsAndAuthorities( $G$ )
6   for step from 1 to  $k$  do // run the algorithm for  $k$  steps
7      $norm = 0$ 
8     for each page  $p$  in  $G$  do // update all authority values first
9        $p.auth = 0$ 
10      for each page  $q$  in  $p.incomingNeighbors$  do //  $p.incomingNeighbors$  is the set
of pages that link to  $p$ 
11         $p.auth += q.hub$ 
12       $norm += square(p.auth)$  // calculate the sum of the squared auth values to
normalise
13       $norm = sqrt(norm)$ 
14      for each page  $p$  in  $G$  do // update the auth scores
15         $p.auth = p.auth / norm$  // normalise the auth values
16       $norm = 0$ 
17      for each page  $p$  in  $G$  do // then update all hub values
18         $p.hub = 0$ 
19        for each page  $r$  in  $p.outgoingNeighbors$  do //  $p.outgoingNeighbors$  is the set
of pages that  $p$  links to
20           $p.hub += r.auth$ 
21         $norm += square(p.hub)$  // calculate the sum of the squared hub values to
normalise
22         $norm = sqrt(norm)$ 
23        for each page  $p$  in  $G$  do // then update all hub values
24           $p.hub = p.hub / norm$  // normalise the hub values
```

The hub and authority values converge in the pseudocode above.

The code below does not converge, because it is necessary to limit the number of steps that the algorithm runs for. One way to get around this, however, would be to normalize the hub and authority values after each "step" by dividing each authority value by the square root of the sum of the squares of all authority values, and dividing each hub value by the square root of the sum of the squares of all hub values. This is what the pseudocode above does.




Non-converging pseudocode [\[edit\]](#)

```
1  $G :=$  set of pages
2 for each page  $p$  in  $G$  do
3    $p.auth = 1$  //  $p.auth$  is the authority score of the page  $p$ 
4    $p.hub = 1$  //  $p.hub$  is the hub score of the page  $p$ 
5 function HubsAndAuthorities( $G$ )
6   for step from 1 to  $k$  do // run the algorithm for  $k$  steps
7     for each page  $p$  in  $G$  do // update all authority values first
8        $p.auth = 0$ 
9       for each page  $q$  in  $p.incomingNeighbors$  do //  $p.incomingNeighbors$  is the set
of pages that link to  $p$ 
10         $p.auth += q.hub$ 
11      for each page  $p$  in  $G$  do // then update all hub values
12         $p.hub = 0$ 
13        for each page  $r$  in  $p.outgoingNeighbors$  do //  $p.outgoingNeighbors$  is the set
of pages that  $p$  links to
14           $p.hub += r.auth$ 
```



References [\[edit\]](#)

1. [^] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze (2008). "Introduction to Information Retrieval" [↗](#). Cambridge University Press. Retrieved 2008-11-09.
2. [^] Kleinberg, Jon (December 1999). "Hubs, Authorities, and Communities" [↗](#). Cornell University. Retrieved 2008-11-09.
3. [^] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh [WTF: The who-to-follow system at Twitter](#) [↗](#), Proceedings of the 22nd international conference on World Wide Web
4. [^] von Ahn, Luis (2008-10-19). "Hubs and Authorities" [↗](#) (PDF). 15-396: *Science of the Web Course Notes*. Carnegie

Mellon University. Retrieved 2015-01-19.

- Kleinberg, Jon (1999). "Authoritative sources in a hyperlinked environment"  (PDF). *Journal of the ACM* **46** (5): 604–632. doi:10.1145/324133.324140 .
- Li, L.; Shang, Y.; Zhang, W. (2002). "Improvement of HITS-based Algorithms on Web Documents" . *Proceedings of the 11th International World Wide Web Conference (WWW 2002)*. Honolulu, HI. ISBN 1-880672-20-0.

External links

- U.S. Patent 6,112,202 
- Create a data search engine from a relational database  Search engine in C# based on HITS

Categories: Link analysis

This page was last modified on 19 June 2015, at 06:48.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

