

3 Searching the Swarm and Measuring Emergence

We think of the World Wide Web (WWW) as an example of a dynamic swarm of web pages providing information unconsciously. More precisely:

- the WWW is a hypertext corpus with enormous complexity
 - it (still) grows very fast (\leadsto dynamics)
 - change of pages' contents, pages appear and disappear (\leadsto dynamics)
 - users with different and often conflicting goals generate hyperlinked content at the most local level
- created by users (one or two page, not the whole web)*

In the following, we describe how to find information relevant to a query in this "swarm" like a pilot fish finding food in a swarm of host fishes.

We distinguish several types of queries in a search query:

1. • **Specific query:** "Does netscape support the JDK 1.1 codesigning API?"

The problem here is that very few pages contain this information.

2. • **Broad-topic query:** "Find information about the Java programming language"

Here the problem is the overabundance of web pages covering the topic – especially for text-based searches.

3. • **Similar-page query:** "Find similar pages like java.sun.com"
- Java is Island, Java is a Coffee! → for car manufacturers → webs like BMW, Volvo, Mercedes → for inspiration*

Here, we don't investigate the corresponding clustering problem, where it is necessary, for example, to separate pages that consider real keys (🔑) from those that are about encrypting (as in cryptography).

Due to the abundance of potential results in a broad-topic query, our goal is to identify the (most) relevant pages, the so-called authorities. However, here the question arises how to determine or measure the authority of a page, or, in other words, how to measure the emergent property of a web page to provide good (best?) information regarding the query (recall the research question "metrics for ... emergence phenomena" from Sec. 1.4).

The following complications must be considered:

- The page where the query string appears most often does not necessarily have to be the page with the highest authority (example: spam pages) \leadsto authority is presumably not an endogenous measure that can be determined based on a page alone.
- The page with highest authority does not even necessarily have to contain the search string (example: audi.de and "car manufacturer").

Relevance of 1 page depends on Relevance of all other competitive pages!

⌘ This leads to the model assumption, that evaluating the link structure between the pages leads to good results because it contains human ratings (see our definition of emergence). However, it must be taken into account that neither navigation links nor advertising links should contribute to the authority of a page.

A first heuristic for answering a broad-topic query consists of outputting those of all pages that contain the search string that are pointed to the most. However, two main problems arise with this heuristic:

clicked most no if time.

- An authority that does not contain the search string will not appear in the result (\leadsto Audi).
- Popular pages that are frequently hyperlinked become authorities on any topic.

It turns out (Kleinberg's idea) that besides the authorities there are also those pages which know the authorities, the so-called **hubs** (a hub is the center part of a bicycle wheel (\odot)), which are in a certain way authorities for authorities. So in the following the goal now is to determine authorities and hubs.

The HITS Algorithm

HITS stands for Hyperlink-Induced Topic Search, or Hypertext-Induced Topic Search and was introduced by J. Kleinberg in 1999.

\rightarrow not defined by Author

- **Given:** a broad-topic query using search string σ .
- **Goal:** Output high priority pages

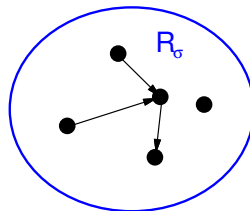
First, we compute a **base graph** S_σ with the following properties: *small graph, we hope we find result here / else we expand.*

- (i) S_σ is relatively small. *(faster comp)*
- (ii) S_σ contains many relevant pages.
- (iii) S_σ contains many of the pages with highest authority.

We start with a simple text-based search that gives us the t "best" pages as a **root graph** R_σ . This root graph only partially satisfies the above conditions:

(start point)

- (i) + (ii) ✓ *Hopefully (as they have string σ)*
 - (iii) ☹️
- "200 pages"*



Experimental investigations of sample root graphs have shown a phenomenon: R_σ has hardly internal edges (in the invention paper of the HITS algorithm, for example, for $t = 200$ just 28 internal links were found whereas $\binom{200}{2} \cdot 2 = 200 \cdot 199 = 39,800$ might be possible).

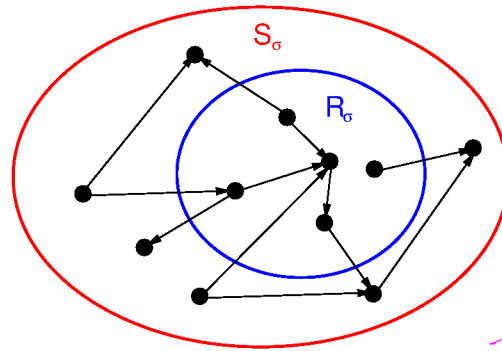
If a page with high authority regarding σ is not included in R_σ , the probability is high that a page from R_σ points to it. For a hub, the reverse is true, i.e., the probability is high that the hub page points to a page in R_σ . *a "good hub"*

We therefore extend R_σ by its immediate outside neighbors in the web graph, limiting the set of new entering edges by a parameter d to keep the size of the resulting **base graph** S_σ relatively small.

\hookrightarrow to avoid explosion of nodes!

how this influences this?

① *How hubs know about authority of other webpages?*



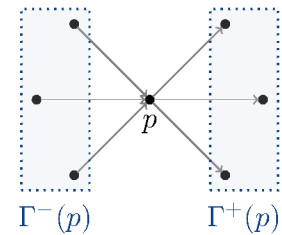
any page can be one or both of Hub/Authority for our query.
 → So each page has two weights → Hub, Authority

Algorithmus 2 generation of S_σ from R_σ by (partial) addition of neighbors

```

 $S_\sigma := R_\sigma$ 
for all  $p \in R_\sigma$  do
   $S_\sigma := S_\sigma \cup \Gamma^+(p)$  (extend  $S_\sigma$  by outgoing links)
  if  $|\Gamma^-(p)| \leq d$  then (if incoming links less than d) (as many pages can link to a page if it has say high authority)
     $S_\sigma := S_\sigma \cup \Gamma^-(p)$ 
  else
     $S_\sigma := S_\sigma \cup$  "select  $d$  randomly chosen pages from  $\Gamma^-(p)$ " {Sampling} (if too many pages is pointing to p)
  end if
end for
  
```

Here $\Gamma^+(p)$ denotes the neighbors of p to which p points (outgoing edges). Similarly, $\Gamma^-(p)$ denotes the neighbors pointing to p (incoming edges). The diagram to the right represents the two sets.



The set $\Gamma^+(p)$ is in general relatively small, whereas $\Gamma^-(p)$ can obviously be huge.

Experiments reported in the invention paper show that for $t = 200$ and $d = 50$ the expected cardinality of the node set of S_σ is about 1000 to 5000.

We also apply two additional heuristics to the base graph S_σ to improve the quality of the search result:

- Delete internal (w.r.t. domain name) links, keep external links → Eliminate navigation links. within website
- Allow only m (4 through 8) links from a domain to a page (avoiding "This page was created using ...") → contact page etc etc.

Now a distinction between popular and actually relevant pages is necessary.

Note: Compared to the whole web graph sorting the pages of S_σ (after applying above heuristics) w.r.t. the in-degree already returns a quite useful result.

The goal now is to compute (as actual numbers) the authorities of the pages in S_σ .

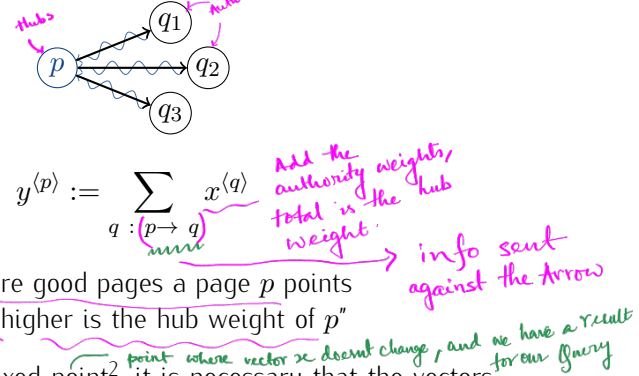
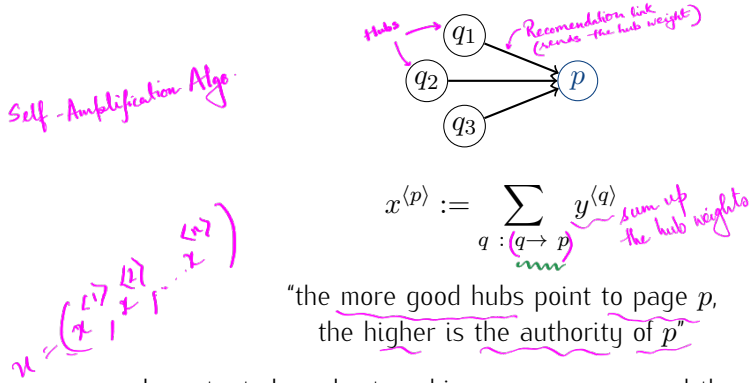
Pages with high authority w.r.t. σ should not only have a large in-degree, but those pages pointing to them should overlap considerably, i.e., should be hubs.

To compute the authorities and hubs, we assign to each side p the authority weight $x^{(p)}$ and the hub weight $y^{(p)}$:
 page p
 $x = \text{vector}(x^{(1)}, x^{(2)}, \dots, x^{(n)})$

- The larger the value of $x^{(p)}$, the more suitable the page p is as an answer to the query.

- The larger the value of $y^{(p)}$, the more "significant" the page p is as a hub.

We note that authorities and hubs reinforce each other. From this we can derive the following update rules for the authority and hub weights, which must be applied repeatedly up to a fixed point:



Important: In order to achieve convergence and thus maintain a fixed point², it is necessary that the vectors x and y are normalized after each recalculation, i.e., we transform x and y to unit vectors after each step, so that holds:

$$\sum_{p \in S_\sigma} (x^{(p)})^2 = 1, \quad \sum_{p \in S_\sigma} (y^{(p)})^2 = 1$$

Reduce the vector x, y to $(0, 1)$ else diverge

This yields the following iterative algorithm for step-wise computation of the authority and lift weights of the sides in S_σ :

Algorithmus 3 $\text{iterate}(k)$

input: k : Number of iterations, S_σ : base graph

output: x : authority weights of the pages in S_σ (as this gives the search result) (also take out hub weights).

$n :=$ number of pages in S_σ

$x := \mathbf{1} \in \mathbb{R}^n$

$y := \mathbf{1} \in \mathbb{R}^n$

for $i := 1$ to k do

$x^{(p)} = \sum_{q: q \rightarrow p} y^{(q)}$

$y^{(p)} = \sum_{q: p \rightarrow q} x^{(q)}$

normalize x, y

end for

We are now interested in whether the vectors x and y computed in $\text{iterate}(k)$ converge for $k \rightarrow \infty$ because obviously the results are "usable" only in this case. For the analysis we use the adjacency matrix A of the graph S_σ :

have a meaning

$$A_{ij} = \begin{cases} 1 & \text{if } i \rightarrow j \text{ is existing edge} \\ 0 & \text{otherwise} \end{cases}$$

Thus, the calculations of x and y can be expressed as follows (T means: transpose the matrix):

$$x := A^T \cdot y$$

$$y := A \cdot x$$

why!

²The direct application of the update rules for the authority and hub weights leads to diverging values, so that the final values would only be reached after an infinite number of iterations

Now it is easy to show that x and y converge for $k \rightarrow \infty$ because:

Without Normalization \rightarrow $y_k = (A \cdot A^T) \cdot y_{k-1} \Rightarrow y_k = (A \cdot A^T)^k \cdot \mathbf{1}$ \rightarrow time steps (like exercise)
 $A \cdot A^T$ is symmetric $\Rightarrow y$ converges to principal eigenvector of $A \cdot A^T$ \rightarrow becoz after infinite steps, the principal eigenvalue, will be the only dominant direction, so x/y converge to Principal Axis.
 and
 $x_k = (A^T \cdot A)^{k-1} \cdot A^T \cdot \mathbf{1}$
 $A^T \cdot A$ is symmetric $\Rightarrow x$ converges to the principal eigenvector v of $(A^T \cdot A)$ times A^T
 from linear algebra and normalization
 A is not symmetric But... $A^T \cdot A$ is symmetric

The principal eigenvector of a matrix M is the eigenvector belonging to the largest eigenvalue of M .

As $A \cdot A^T$ and $A^T \cdot A$ are symmetric, all eigenvalues are real-valued, and, hence, the principal eigenvector has only real entries. (as over PageRank algo, there λ 's can be complex)

These entries of x now are the authority weights of the web pages. They can be sorted, and the most relevant page is the one with the highest authority weight.


If new pages show up in the WWW, due to the interplay of the pages, the authority weights of the pages may change.

Experiments suggest that $k = 20$ for expander graphs³ is also sufficient to obtain approximate convergence.

Kleinberg's paper:

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46 (1999) 604–632. doi:10.1145/324133.324140

So Auth / Hub weights is an Emergent Property

Expander graphs \rightarrow counter eg: Mesh.
 iff, $\Gamma(u) = \text{at least } \frac{1}{2}(u)$


Most graphs are expander graphs. Any randomly generated graph with Prob=1 will be an expander graph.

³An expander graph is informally a graph where every relatively large subset of nodes has many neighbors outside this set of nodes. The World Wide Web is an expander graph, and thus S_σ as a adequately large subgraph is most likely an expander as well.

