ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΙ ΑΝΑΖΗΤΗΣΗ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

Παροράματα από το Πανεπιστήμιο της Στουγκάρδης

Introduction to

Information Retrieval

Hinrich Schütze and Christina Lioma

Lecture 21: Link Analysis

Outline

1. Recap
2. Anchor Text
3. Citation Analysis
4. PageRank
5. HITS: Hubs & Authorities

Outline

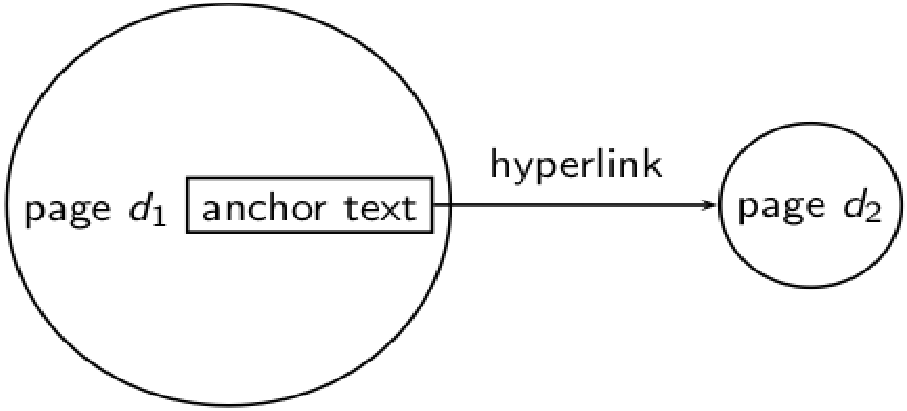
1. Recap
2. Anchor Text
3. Citation Analysis
4. PageRank
5. HITS: Hubs & Authorities

Outline

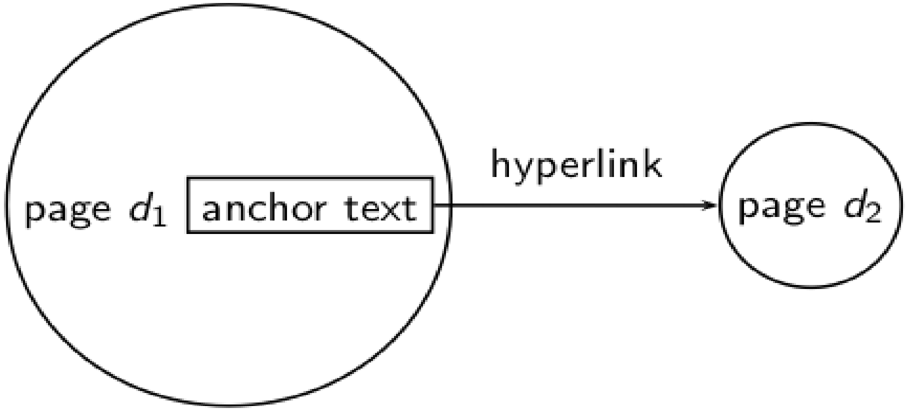
1. Recap
2. Anchor Text
3. Citation Analysis
4. PageRank
5. HITS: Hubs & Authorities

The web as a directed graph

The web as a directed graph

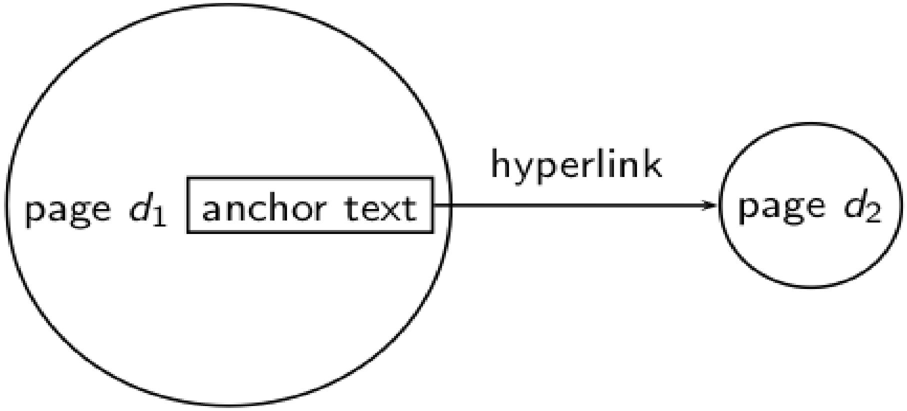


The web as a directed graph



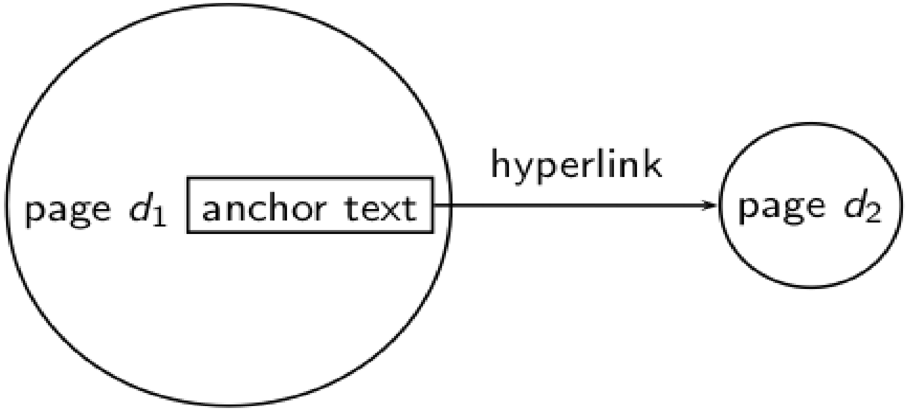
- Assumption 1: A hyperlink is a quality signal.

The web as a directed graph



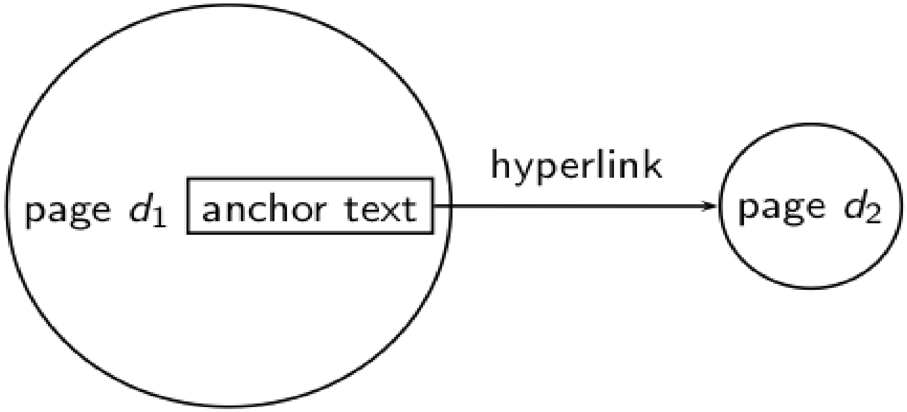
- Assumption 1: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.

The web as a directed graph



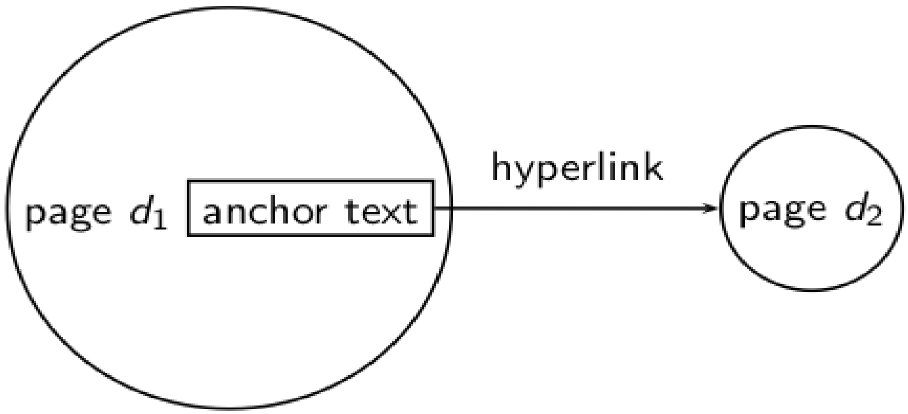
- Assumption 1: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.
- Assumption 2: The anchor text describes the content of d_2.

The web as a directed graph



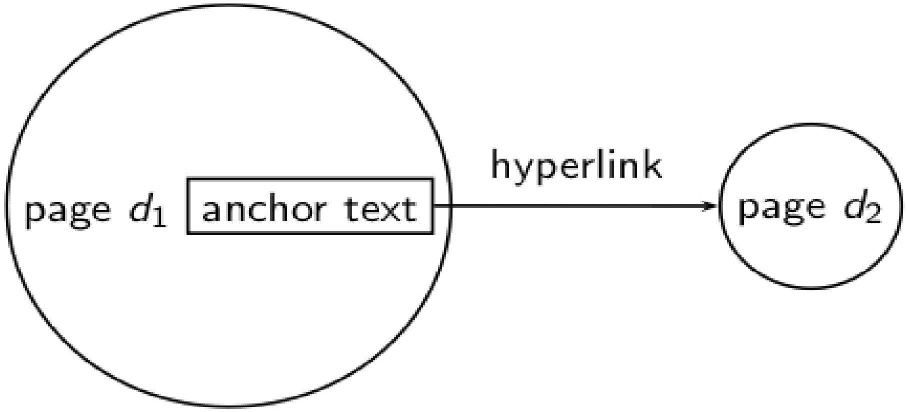
- Assumption 1: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.
- Assumption 2: The anchor text describes the content of d_2.
  - We use anchor text somewhat loosely here for: the text surrounding the hyperlink .

The web as a directed graph



- Assumption 1: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.
- Assumption 2: The anchor text describes the content of d_2.
  - We use anchor text somewhat loosely here for: the text surrounding the hyperlink .
  - Example: "You can find cheap cars <a href =http://…>here . "

The web as a directed graph



- Assumption 1: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.
- Assumption 2: The anchor text describes the content of d_2.
  - We use anchor text somewhat loosely here for: the text surrounding the hyperlink .
  - Example: “You can find cheap cars <a href =http://…>here . ”
  - Anchor text: “You can find cheap here”

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page!

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page!
  - … if IBM home page is mostly graphics

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

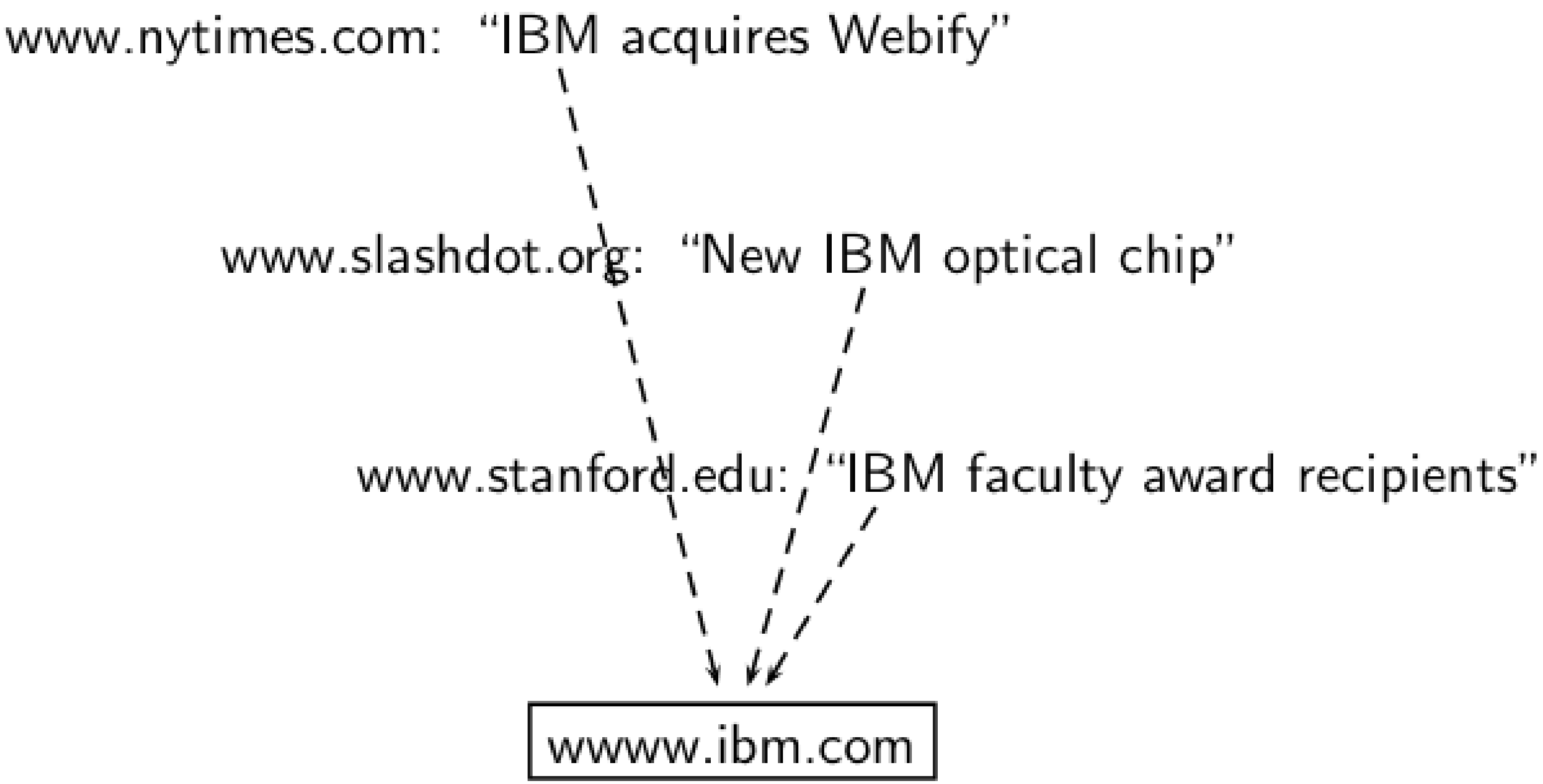
- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page!
  - ... if IBM home page is mostly graphics
- Searching on [anchor text → d_2] is better for the query *IBM*.

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page!
  - … if IBM home page is mostly graphics
- Searching on [anchor text → d_2] is better for the query *IBM*.
  - In this representation, the page with most occurences of *IBM* is www.ibm.com

Anchor text containing *IBM* pointing to www.ibm.com

Anchor text containing *IBM* pointing to www.ibm.com



Indexing anchor text

Indexing anchor text

- Thus: Anchor text is often a better description of a page's content than the page itself.

Indexing anchor text

- Thus: Anchor text is often a better description of a page's content than the page itself.
- Anchor text can be weighted more highly than document text.

  (based on Assumption 1&2)

Exercise: Assumptions underlying PageRank

Exercise: Assumptions underlying PageRank

- Assumption 1: A link on the web is a quality signal – the author of the link thinks that the linked-to page is high-quality.

Exercise: Assumptions underlying PageRank

- Assumption 1: A link on the web is a quality signal – the author of the link thinks that the linked-to page is high-quality.
- Assumption 2: The anchor text describes the content of the linked-to page.

Exercise: Assumptions underlying PageRank

- Assumption 1: A link on the web is a quality signal – the author of the link thinks that the linked-to page is high-quality.
- Assumption 2: The anchor text describes the content of the linked-to page.
- Is assumption 1 true in general?

Exercise: Assumptions underlying PageRank

- Assumption 1: A link on the web is a quality signal – the author of the link thinks that the linked-to page is high-quality.
- Assumption 2: The anchor text describes the content of the linked-to page.
- Is assumption 1 true in general?
- Is assumption 2 true in general?

Google bombs

Google bombs

- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.

Google bombs

- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.

Google bombs

- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.
- Still some remnants: [dangerous cult] on Google, Bing, Yahoo

Google bombs

- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.
- Still some remnants: [dangerous cult] on Google, Bing, Yahoo
  - Coordinated link creation by those who dislike the Church of Scientology

Google bombs

- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.
- Still some remnants: [dangerous cult] on Google, Bing, Yahoo
  - Coordinated link creation by those who dislike the Church of Scientology
- Defused Google bombs: [dumb motherf…], [who is a failure?], [evil empire]

Outline

1. Recap
2. Anchor Text
3. Citation Analysis
4. PageRank
5. HITS: Hubs & Authorities

Origins of PageRank: Citation analysis (1)

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- We can view "Miller (2001)" as a hyperlink linking two scientific articles.

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- We can view "Miller (2001)" as a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- We can view "Miller (2001)" as a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- We can view "Miller (2001)" as a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.
  - This is called cocitation similarity.

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- We can view "Miller (2001)" as a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.
  - This is called cocitation similarity.
  - Cocitation similarity on the web: Google's "find pages like this" or "Similar" feature.

Origins of PageRank: Citation analysis (2)

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
  - ... mainly because of link spam.

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
  - ... mainly because of link spam.
- Better measure: weighted citation frequency or citation rank

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
  - ... mainly because of link spam.
- Better measure: weighted citation frequency or citation rank
  - An article's vote is weighted according to its citation impact.

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
  - ... mainly because of link spam.
- Better measure: weighted citation frequency or citation rank
  - An article’s vote is weighted according to its citation impact.
  - Circular? No: can be formalized in a well-defined way.

Origins of PageRank: Citation analysis (3)

Origins of PageRank: Citation analysis (3)

- Better measure: weighted citation frequency or citation rank.

Origins of PageRank: Citation analysis (3)

- Better measure: weighted citation frequency or citation rank.
- This is basically PageRank.

Origins of PageRank: Citation analysis (3)

- Better measure: weighted citation frequency or citation rank.
- This is basically PageRank.
- PageRank was invented in the context of citation analysis by Pinsker and Narin in the 1960s.

Origins of PageRank: Citation analysis (3)

- Better measure: weighted citation frequency or citation rank.
- This is basically PageRank.
- PageRank was invented in the context of citation analysis by Pinsker and Narin in the 1960s.
- Citation analysis is a big deal: The budget and salary of this lecturer are / will be determined by the impact of his publications!

Origins of PageRank: Summary

Origins of PageRank: Summary

- We can use the same formal representation for

Origins of PageRank: Summary

- We can use the same formal representation for
  - citations in the scientific literature

Origins of PageRank: Summary

- We can use the same formal representation for
  - citations in the scientific literature
  - hyperlinks on the web

Origins of PageRank: Summary

- We can use the same formal representation for
  - citations in the scientific literature
  - hyperlinks on the web
- Appropriately weighted citation frequency is an excellent measure of quality ...

Origins of PageRank: Summary

- We can use the same formal representation for
  - citations in the scientific literature
  - hyperlinks on the web
- Appropriately weighted citation frequency is an excellent measure of quality ...
  - ... both for web pages and for scientific publications.

Origins of PageRank: Summary

- We can use the same formal representation for
  - citations in the scientific literature
  - hyperlinks on the web
- Appropriately weighted citation frequency is an excellent measure of quality ...
  - ... both for web pages and for scientific publications.
- Next: PageRank algorithm for computing weighted citation frequency on the web.

Outline

1. Recap
2. Anchor Text
3. Citation Analysis
4. PageRank
5. HITS: Hubs & Authorities

Model behind PageRank: Random walk

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web
  - Start at a random page

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state, each page has a long-term visit rate.

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state, each page has a long-term visit rate.
- This long-term visit rate is the page's PageRank.

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state, each page has a long-term visit rate.
- This long-term visit rate is the page's PageRank.
- PageRank = long-term visit rate = steady state probability.

Formalization of random walk: Markov chains

Formalization of random walk: Markov chains

- A Markov chain consists of N states, plus an $N \times N$ transition probability matrix P.

Formalization of random walk: Markov chains

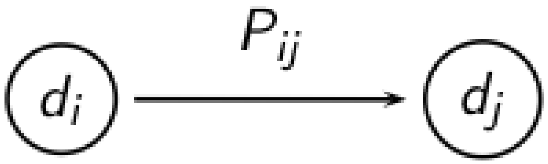
- A Markov chain consists of N states, plus an $N \times N$ transition probability matrix P.
- state = page

Formalization of random walk: Markov chains

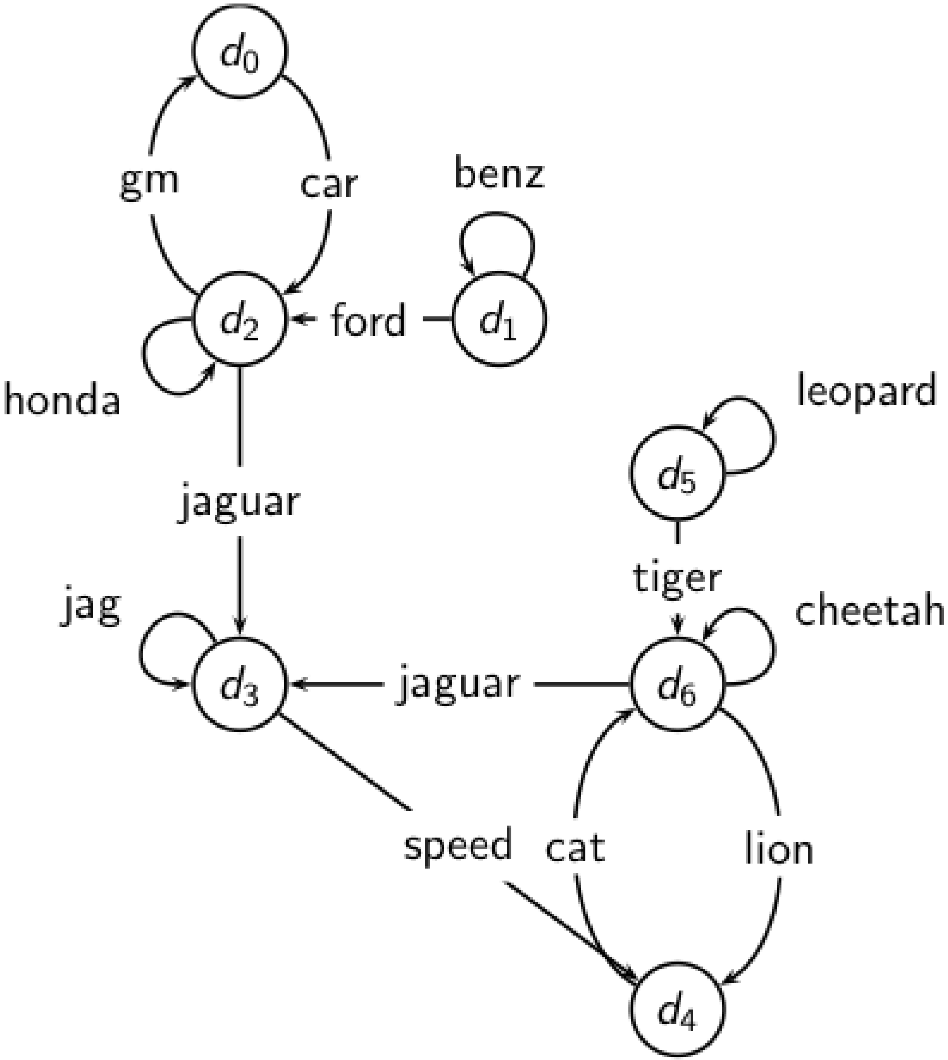
- A Markov chain consists of N states, plus an $N \times N$ transition probability matrix P.
- state = page
- At each step, we are on exactly one of the pages.

Formalization of random walk: Markov chains

- A Markov chain consists of N states, plus an $N \times N$ transition probability matrix P.
- state = page
- At each step, we are on exactly one of the pages.
- For $1 \leq i, j \geq N$, the matrix entry P_{ij} tells us the probability of j being the next page, given we are currently on page i.
- Clearly, for all i, $\sum_{j=1}^{N} P_{ij} = 1$



Example web graph



Link matrix for example

Link matrix for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	1	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	1	1	0	1

Transition probability matrix P for example

Transition probability matrix P for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

Long-term visit rate

Long-term visit rate

- Recall: PageRank = long-term visit rate.

Long-term visit rate

- Recall: PageRank = long-term visit rate.
- Long-term visit rate of page *d* is the probability that a web surfer is at page *d* at a given point in time.

Long-term visit rate

- Recall: PageRank = long-term visit rate.
- Long-term visit rate of page *d* is the probability that a web surfer is at page *d* at a given point in time.
- Next: what properties must hold of the web graph for the long-term visit rate to be well defined?

Long-term visit rate

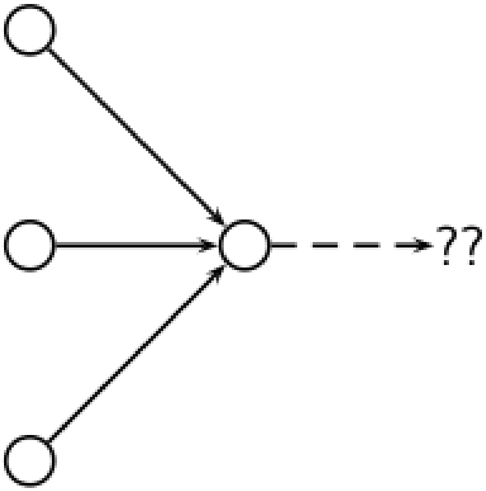
- Recall: PageRank = long-term visit rate.
- Long-term visit rate of page *d* is the probability that a web surfer is at page *d* at a given point in time.
- Next: what properties must hold of the web graph for the long-term visit rate to be well defined?
- The web graph must correspond to an ergodic Markov chain.

Long-term visit rate

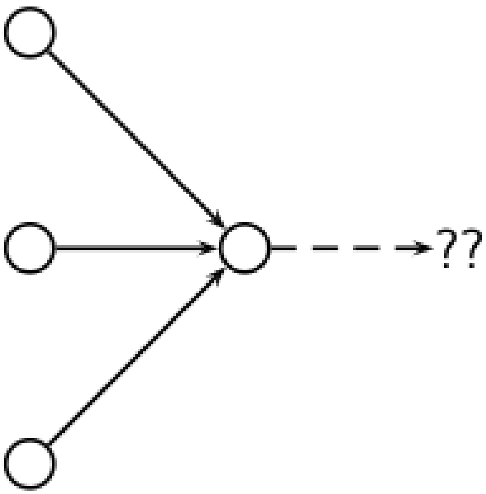
- Recall: PageRank = long-term visit rate.
- Long-term visit rate of page *d* is the probability that a web surfer is at page *d* at a given point in time.
- Next: what properties must hold of the web graph for the long-term visit rate to be well defined?
- The web graph must correspond to an ergodic Markov chain.
- First a special case: The web graph must not contain dead ends.

Dead ends

Dead ends

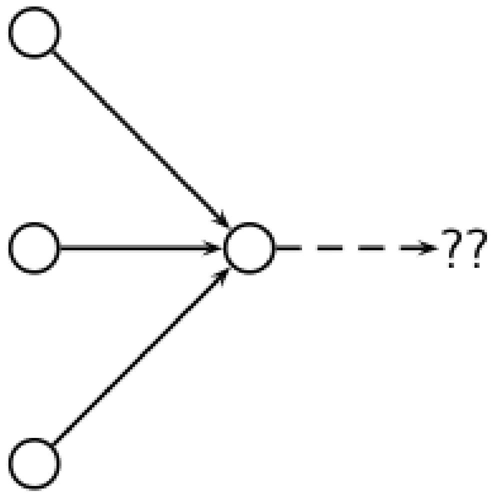


Dead ends



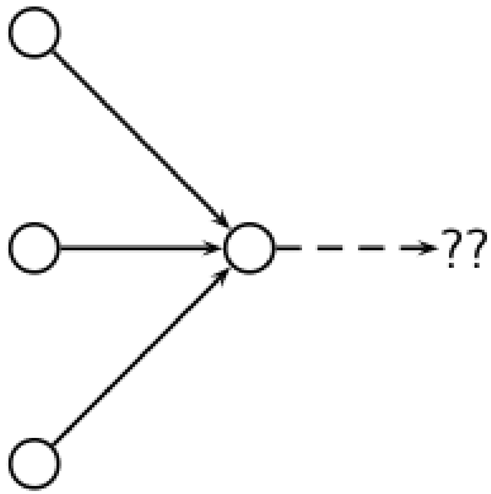
- The web is full of dead ends.

Dead ends



- The web is full of dead ends.
- Random walk can get stuck in dead ends.

Dead ends



- The web is full of dead ends.
- Random walk can get stuck in dead ends.
- If there are dead ends, long-term visit rates are not well-defined (or non-sensical).

Teleporting – to get us of dead ends

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. $1/N$.

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. $1/N$.
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of $0.1/N$).

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. $1/N$.
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of $0.1/N$).
- With remaining probability (90%), go out on a random hyperlink.

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. $1/N$.
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of $0.1/N$).
- With remaining probability (90%), go out on a random hyperlink.
    - For example, if the page has 4 outgoing links: randomly choose one with probability (1-0.10)/4=0.225

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. 1/ *N* .
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of 0.1/*N*).
- With remaining probability (90%), go out on a random hyperlink.
  - For example, if the page has 4 outgoing links: randomly choose one with probability (1-0.10)/4=0.225
- 10% is a parameter, the teleportation rate.

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. 1/ N .
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of 0.1/N).
- With remaining probability (90%), go out on a random hyperlink.
  - For example, if the page has 4 outgoing links: randomly choose one with probability (1-0.10)/4=0.225
- 10% is a parameter, the teleportation rate.
- Note: “jumping” from dead end is independent of teleportation rate.

Result of teleporting

Result of teleporting

- With teleporting, we cannot get stuck in a dead end.

Result of teleporting

- With teleporting, we cannot get stuck in a dead end.
- But even without dead ends, a graph may not have well-defined long-term visit rates.

Result of teleporting

- With teleporting, we cannot get stuck in a dead end.
- But even without dead ends, a graph may not have well-defined long-term visit rates.
- More generally, we require that the Markov chain be ergodic.

Ergodic Markov chains

- A Markov chain is ergodic if it is irreducible and aperiodic.

Ergodic Markov chains

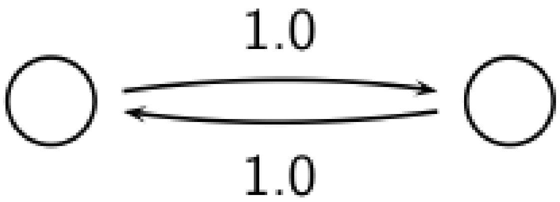
- A Markov chain is ergodic if it is irreducible and aperiodic.
- Irreducibility. Roughly: there is a path from any other page.

Ergodic Markov chains

- A Markov chain is ergodic if it is irreducible and aperiodic.
- Irreducibility. Roughly: there is a path from any other page.
- Aperiodicity. Roughly: The pages cannot be partitioned such that the random walker visits the partitions sequentially.

Ergodic Markov chains

- A Markov chain is ergodic if it is irreducible and aperiodic.
- Irreducibility. Roughly: there is a path from any other page.
- Aperiodicity. Roughly: The pages cannot be partitioned such that the random walker visits the partitions sequentially.
- A non-ergodic Markov chain:



Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.

Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.

Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.
- Over a long time period, we visit each state in proportion to this rate.

Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.
- Over a long time period, we visit each state in proportion to this rate.
- It doesn't matter where we start.

Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.
- Over a long time period, we visit each state in proportion to this rate.
- It doesn’t matter where we start.
- Teleporting makes the web graph ergodic.

Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.
- Over a long time period, we visit each state in proportion to this rate.
- It doesn't matter where we start.
- Teleporting makes the web graph ergodic.
- $\Longrightarrow$Web-graph+teleporting has a steady-state probability distribution.

Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.
- Over a long time period, we visit each state in proportion to this rate.
- It doesn’t matter where we start.
- Teleporting makes the web graph ergodic.
- $\Longrightarrow$Web-graph+teleporting has a steady-state probability distribution.
- $\Longrightarrow$Each page in the web-graph+teleporting has a PageRank.

Formalization of “visit”: Probability vector

Formalization of "visit": Probability vector

- A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.

Formalization of "visit": Probability vector

- A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.
- Example:

(	0	0	0	…	1	…	0	0	0	)
	1	2	3	…	*i*	…	N-2	N-1	N	

Formalization of "visit": Probability vector

- A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.
- Example:

(	0	0	0	…	1	…	0	0	0)
	1	2	3	…	*i*	…	N-2	N-1	N

- More generally: the random walk is on the page *i* with probability x_i.

Formalization of "visit": Probability vector

- A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.
- Example:

(	0	0	0	…	1	…	0	0	0	)
	1	2	3	…	*i*	…	N-2	N-1	N	

- More generally: the random walk is on the page *i* with probability x_i.
- Example:

(	0.05	0.01	0.0	…	0.2	…	0.01	0.05	0.03	)
	1	2	3	…	*i*	…	N-2	N-1	N	

Formalization of "visit": Probability vector

- A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.
- Example:

(	0	0	0	…	1	…	0	0	0	)
	1	2	3	…	*i*	…	N-2	N-1	N	

- More generally: the random walk is on the page *i* with probability x_i.
- Example:

(	0.05	0.01	0.0	…	0.2	…	0.01	0.05	0.03	)
	1	2	3	…	*i*	…	N-2	N-1	N	

- $\Sigma x_i = 1$

Change in probability vector

- If the probability vector is $\vec{x} = (x_1, \ldots, x_N)$, at this step, what is it at the next step?

Change in probability vector

- If the probability vector is $\vec{x} = (x_1, ..., x_N)$, at this step, what is it at the next step?
- Recall that row *i* of the transition probability matrix *P* tells us where we go next from state *i*.

Change in probability vector

- If the probability vector is $\vec{x} = (x_1, ..., x_N)$, at this step, what is it at the next step?
- Recall that row *i* of the transition probability matrix *P* tells us where we go next from state *i*.
- So from $\vec{x}$, our next state is distributed as $\vec{x}P$.

Steady state in vector notation

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ of probabilities.

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ of probabilities.
- (We use $\vec{\pi}$ to distinguish it from the notation for the probability vector $\vec{x}$.)

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ of probabilities.
- (We use $\vec{\pi}$ to distinguish it from the notation for the probability vector $\vec{x}$.)
- π is the long-term visit rate (or PageRank) of page *i*.

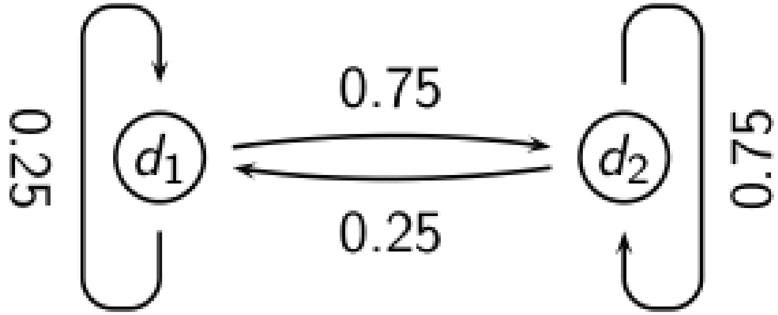
Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ of probabilities.
- (We use $\vec{\pi}$ to distinguish it from the notation for the probability vector $\vec{x}$.)
- π is the long-term visit rate (or PageRank) of page *i*.
- So we can think of PageRank as a very long vector – one entry per page.

Steady-state distribution: Example

Steady-state distribution: Example

- What is the PageRank / steady state in this example?



Steady-state distribution: Example

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$	$P_{12} = 0.75$
			$P_{21} = 0.25$	$P_{22} = 0.75$
t_0	0.25	0.75		
t_1				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$ $P_{21} = 0.25$	$P_{12} = 0.75$ $P_{22} = 0.75$
t_0	0.25	0.75	0.25	0.75
t_1				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$ $P_{21} = 0.25$	$P_{12} = 0.75$ $P_{22} = 0.75$
t_0	0.25	0.75	0.25	0.75
t_1	0.25	0.75		

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$ $P_{21} = 0.25$	$P_{12} = 0.75$ $P_{22} = 0.75$
t_0	0.25	0.75	0.25	0.75
t_1	0.25	0.75	(convergence)	

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

How do we compute the steady state vector?

How do we compute the steady state vector?

- In other words: how do we compute PageRank?

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi}\, P$

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- … and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.
- $\vec{\pi}$ is the principal left eigenvector for P ...

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.
- $\vec{\pi}$ is the principal left eigenvector for P ...
- ... that is, $\vec{\pi}$ is the left eigenvector with the largest eigenvalue.

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.
- $\vec{\pi}$ is the principal left eigenvector for P ...
- ... that is, $\vec{\pi}$ is the left eigenvector with the largest eigenvalue.
- All transition probability matrices have largest eigenvalue 1.

One way of computing the PageRank $\vec{\pi}$

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After *k* steps, we're at $\vec{x}P^k$.

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply $\vec{x}$ by increasing powers of P until convergence.

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After *k* steps, we're at $\vec{x}P^k$.
- Algorithm: multiply $\vec{x}$ by increasing powers of *P* until convergence.
- This is called the power method.

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply $\vec{x}$ by increasing powers of P until convergence.
- This is called the power method.
- Recall: regardless of where we start, we eventually reach the steady state $\vec{\pi}$.

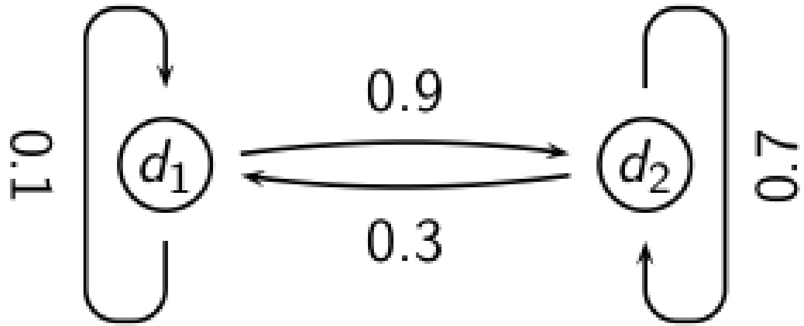
One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After *k* steps, we're at $\vec{x}P^k$.
- Algorithm: multiply $\vec{x}$ by increasing powers of *P* until convergence.
- This is called the power method.
- Recall: regardless of where we start, we eventually reach the steady state $\vec{\pi}$.
- Thus: we will eventually (in asymptotia) reach the steady state.

Power method: Example

Power method: Example

- What is the PageRank / steady state in this example?



Computing PageRank: Power Example

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1			$= \vec{x}P$
t_1					$= \vec{x}P^2$
t_2					$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1					$= \vec{x}P^2$
t_2					$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7			$= \vec{x}P^2$
t_2					$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2					$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76			$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748			$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
				. . .	. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
			. . .		. . .
t_∞	0.25	0.75			$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
			. . .		. . .
t_∞	0.25	0.75	0.25	0.75	$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
				. . .	. . .
t_∞	0.25	0.75	0.25	0.75	$= \vec{x}P^\infty$

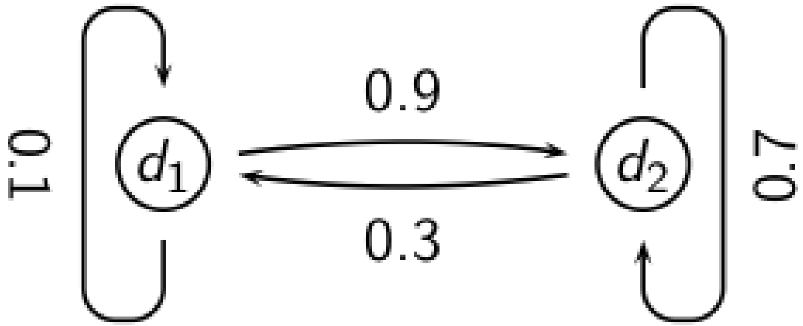
PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Power method: Example

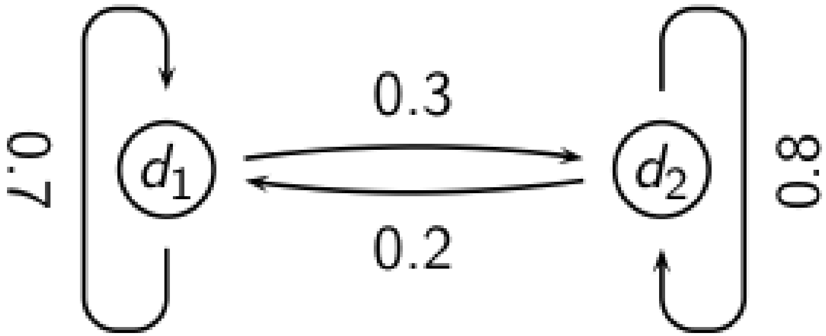
- What is the PageRank / steady state in this example?



- The steady state distribution (= the PageRanks) in this example are 0.25 for d_1 and 0.75 for d_2.

Exercise: Compute PageRank using power method

Exercise: Compute PageRank using power method



Solution

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1		
t_1				
t_2				
t_3				
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1				
t_2				
t_3				
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8		
t_2				
t_3				
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2				
t_3				
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7		
t_3				
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3				
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65		
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
			. . .	
t_∞				

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
			. . .	
t_∞	0.4	0.6		

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
			. . .	
t_∞	0.4	0.6	0.4	0.6

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

PageRank summary

PageRank summary

- Preprocessing

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$
  - $\vec{\pi}_i$ is the PageRank of page *i*.

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$
  - $\vec{\pi}_i$ is the PageRank of page *i*.
- Query processing

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$
  - $\vec{\pi}_i$ is the PageRank of page *i*.
- Query processing
  - Retrieve pages satisfying the query

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$
  - $\vec{\pi}_i$ is the PageRank of page *i*.
- Query processing
  - Retrieve pages satisfying the query
  - Rank them by their PageRank

PageRank summary

- Preprocessing
  - Given graph of links, build matrix *P*
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$
  - $\vec{\pi}_i$ is the PageRank of page *i*.
- Query processing
  - Retrieve pages satisfying the query
  - Rank them by their PageRank
  - Return reranked list to the user

PageRank issues

PageRank issues

- Real surfers are not random surfers.

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it’s good enough as a model for our purposes.

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
  - Consider the query [video service].

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
  - Consider the query [video service].
  - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
  - Consider the query [video service].
  - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.
  - If we rank all Boolean hits according to PageRank, then the Yahoo home page would be top-ranked.

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it’s good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
  - Consider the query [video service].
  - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.
  - If we rank all Boolean hits according to PageRank, then the Yahoo home page would be top-ranked.
  - Clearly not desireble.

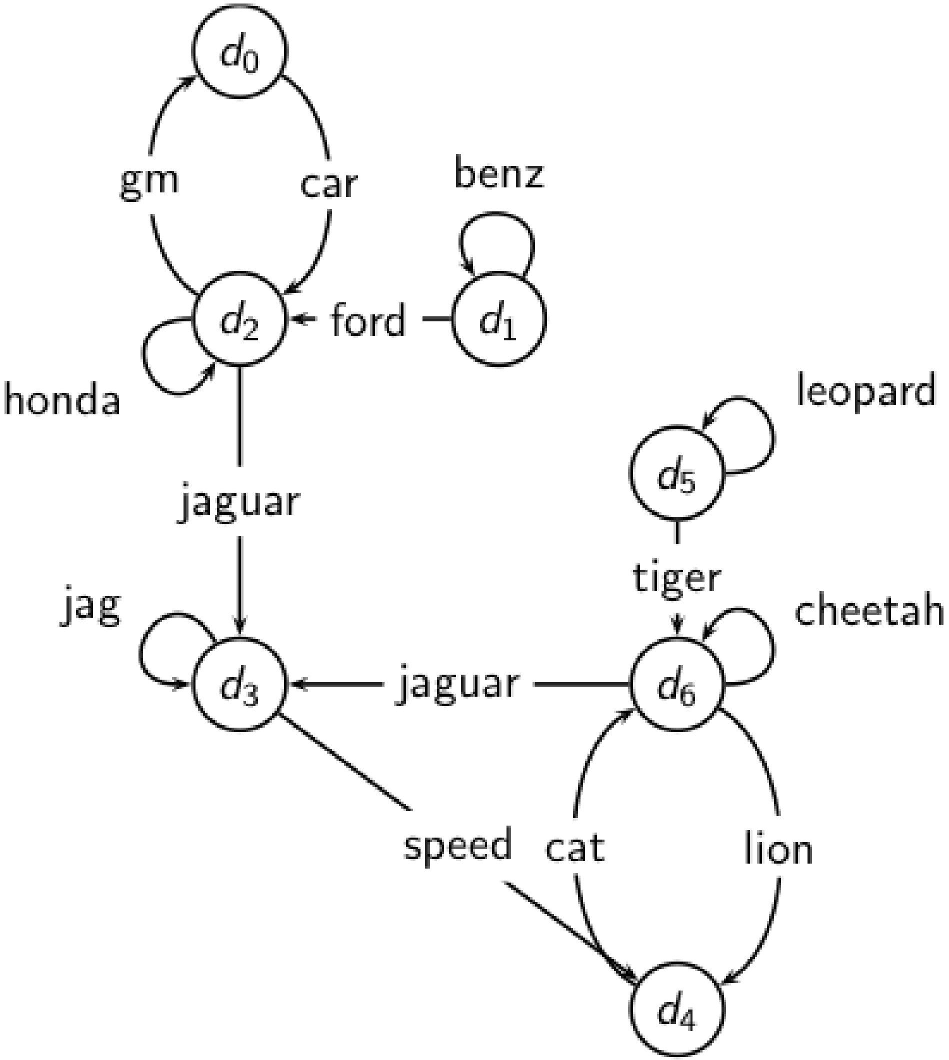
PageRank issues

- In practice: rank according to weighted combination of raw text match, anchor text match, PageRank & other factors.

PageRank issues

- In practice: rank according to weighted combination of raw text match, anchor text match, PageRank & other factors.
- → see lecture on Learning to Rank.

Example web graph



Transition (probability) matrix

Transition (probability) matrix

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

Transition matrix with teleporting

Transition matrix with teleporting

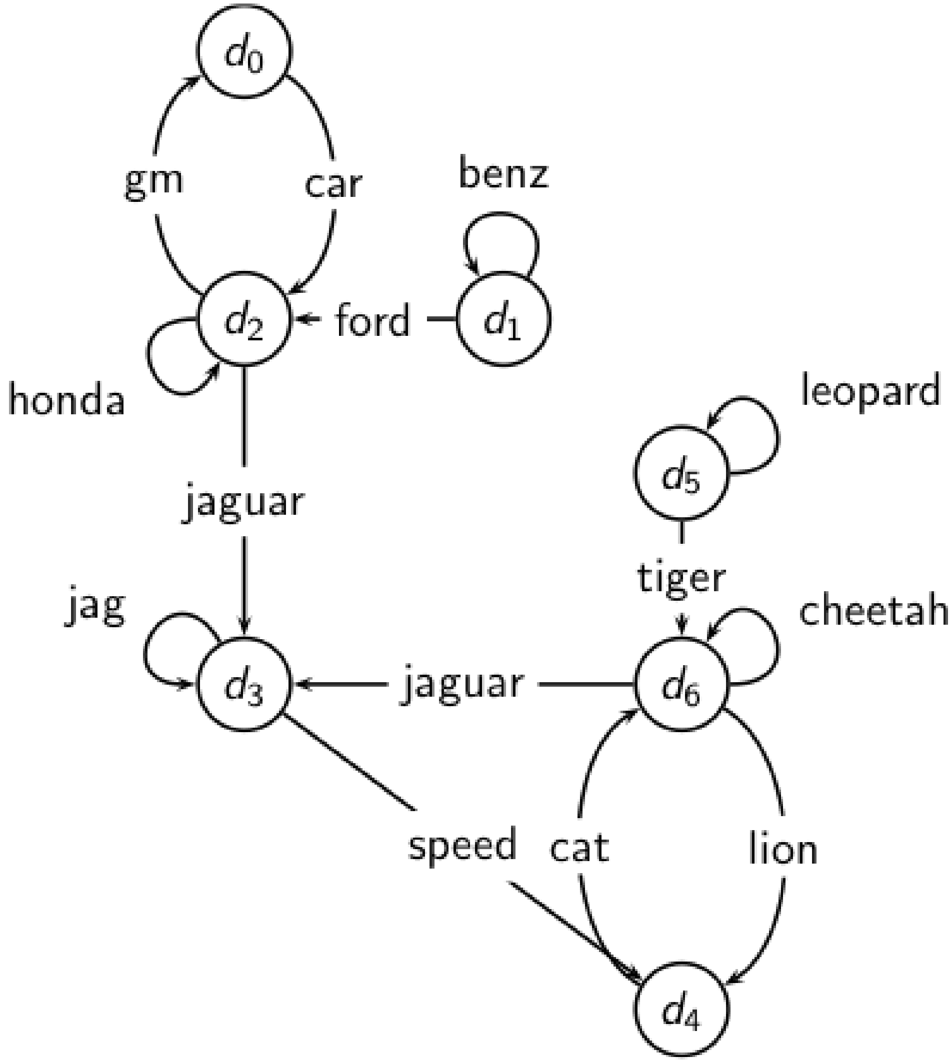
	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.02	0.02	0.88	0.02	0.02	0.02	0.02
d_1	0.02	0.45	0.45	0.02	0.02	0.02	0.02
d_2	0.31	0.02	0.31	0.31	0.02	0.02	0.02
d_3	0.02	0.02	0.02	0.45	0.45	0.02	0.02
d_4	0.02	0.02	0.02	0.02	0.02	0.02	0.88
d_5	0.02	0.02	0.02	0.02	0.02	0.45	0.45
d_6	0.02	0.02	0.02	0.31	0.31	0.02	0.31

Power method vectors $\vec{x}P^k$

Power method vectors $\vec{x}P^k$

	$\vec{x}$	$\vec{x}P^1$	$\vec{x}P^2$	$\vec{x}P^3$	$\vec{x}P^4$	$\vec{x}P^5$	$\vec{x}P^6$	$\vec{x}P^7$	$\vec{x}P^8$	$\vec{x}P^9$	$\vec{x}P^{10}$	$\vec{x}P^{11}$	$\vec{x}P^{12}$	$\vec{x}P^{13}$
d_0	0.14	0.06	0.09	0.07	0.07	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05
d_1	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_2	0.14	0.25	0.18	0.17	0.15	0.14	0.13	0.12	0.12	0.12	0.12	0.11	0.11	0.11
d_3	0.14	0.16	0.23	0.24	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.25	0.25	0.25
d_4	0.14	0.12	0.16	0.19	0.19	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
d_5	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_6	0.14	0.25	0.23	0.25	0.27	0.28	0.29	0.29	0.30	0.30	0.30	0.30	0.31	0.31

Example web graph



	PageRank
d_0	0.05
d_1	0.04
d_2	0.11
d_3	0.25
d_4	0.21
d_5	0.04
d_6	0.31

How important is PageRank?

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
  - Rumor has it that PageRank in his original form (as presented here) now has a negligible impact on ranking!

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
  - Rumor has it that PageRank in his original form (as presented here) now has a negligible impact on ranking!
  - However, variants of a page's PageRank are still an essential part of ranking.

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
  - Rumor has it that PageRank in his original form (as presented here) now has a negligible impact on ranking!
  - However, variants of a page's PageRank are still an essential part of ranking.
  - Adressing link spam is difficult and crucial.

Outline

Hits – Hyperlink-Induced Topic Search

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.
  - E.g, for query [chicago bulls]: Bob's list of recommended resources on the Chicago Bulls sports team

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.
  - E.g, for query [chicago bulls]: Bob's list of recommended resources on the Chicago Bulls sports team
- Relevance type 2: Authorities. An authority page is a direct answer to the information need.

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.
  - E.g, for query [chicago bulls]: Bob's list of recommended resources on the Chicago Bulls sports team
- Relevance type 2: Authorities. An authority page is a direct answer to the information need.
  - The home page of the Chicago Bulls sports team

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.
  - E.g, for query [chicago bulls]: Bob's list of recommended resources on the Chicago Bulls sports team
- Relevance type 2: Authorities. An authority page is a direct answer to the information need.
  - The home page of the Chicago Bulls sports team
  - By definition: Links to authority pages occur repeatedly on hub pages.

Hits – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.
  - E.g, for query [chicago bulls]: Bob's list of recommended resources on the Chicago Bulls sports team
- Relevance type 2: Authorities. An authority page is a direct answer to the information need.
  - The home page of the Chicago Bulls sports team
  - By definition: Links to authority pages occur repeatedly on hub pages.
- Most approaches to search (including PageRank ranking) don't make the distinction between these two very different types of relevance.

Hubs and authorities : Definition

Hubs and authorities : Definition

- A good hub page for a topic links to many authority pages for that topic.

Hubs and authorities : Definition

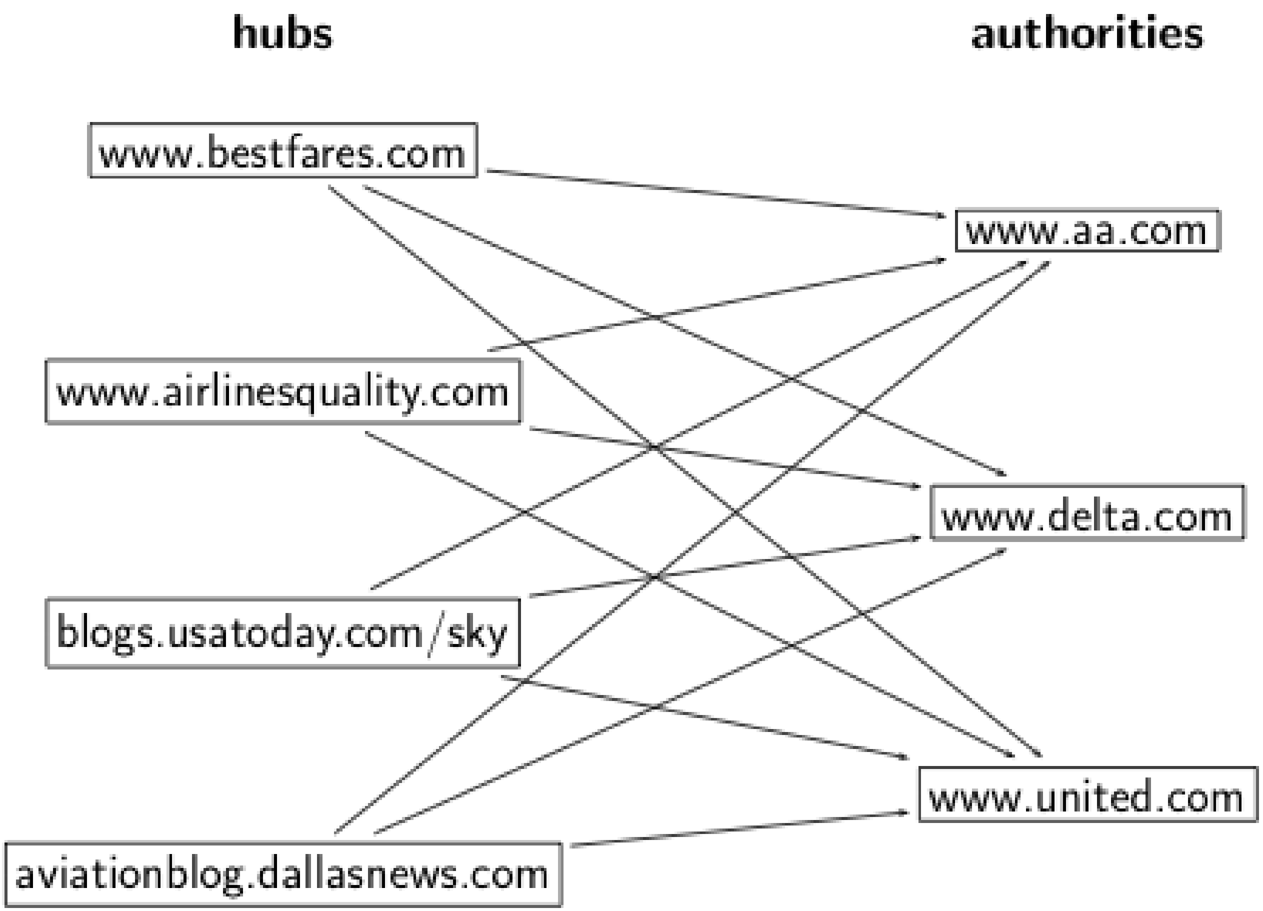
- A good hub page for a topic links to many authority pages for that topic.
- A good authority page for a topic is linked to by many hub pages for that topic.

Hubs and authorities : Definition

- A good hub page for a topic links to many authority pages for that topic.
- A good authority page for a topic is linked to by many hub pages for that topic.
- Circular definition – we will turn this into an iterative computation.

Example for hubs and authorities

Example for hubs and authorities



How to compute hub and authority scores

How to compute hub and authority scores

- Do a regular web search first

How to compute hub and authority scores

- Do a regular web search first
- Call the search result the root set

How to compute hub and authority scores

- Do a regular web search first
- Call the search result the root set
- Find all pages that are linked to or link to pages in the root set

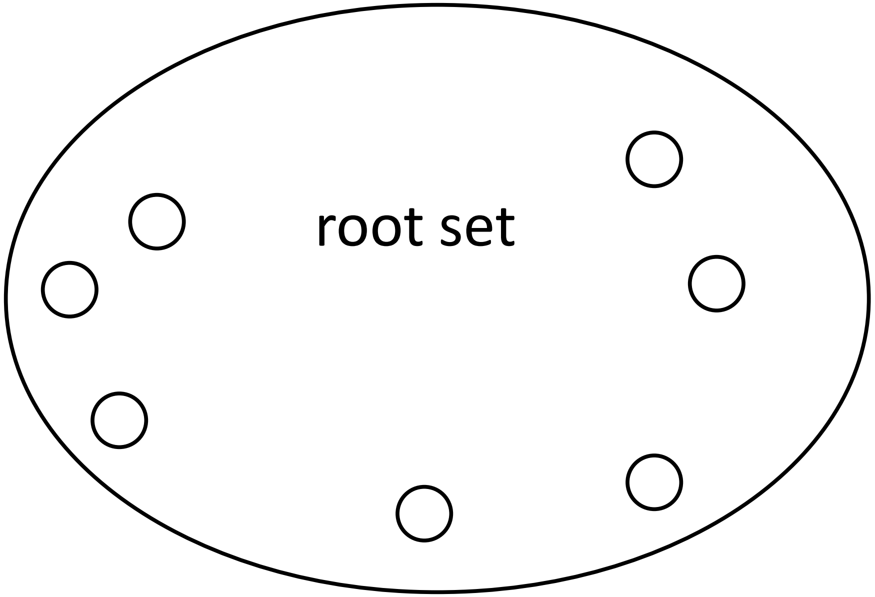
How to compute hub and authority scores

- Do a regular web search first
- Call the search result the root set
- Find all pages that are linked to or link to pages in the root set
- Call first larger set the base set

How to compute hub and authority scores

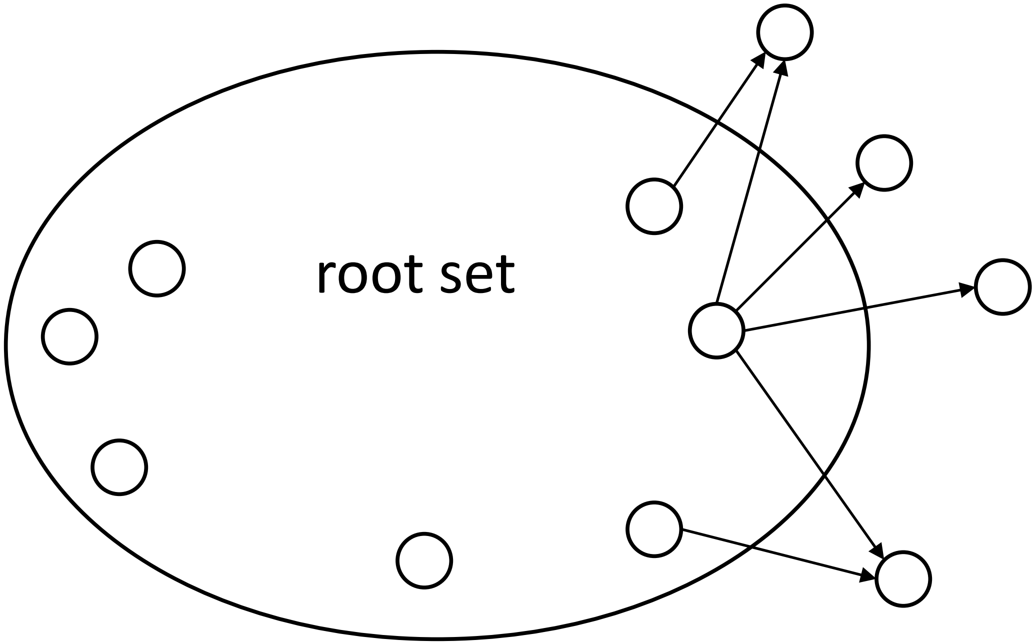
- Do a regular web search first
- Call the search result the root set
- Find all pages that are linked to or link to pages in the root set
- Call first larger set the base set
- Finally, compute hubs and authorities for the base set (which we'll view as a small web graph)

Root set and base set (1)



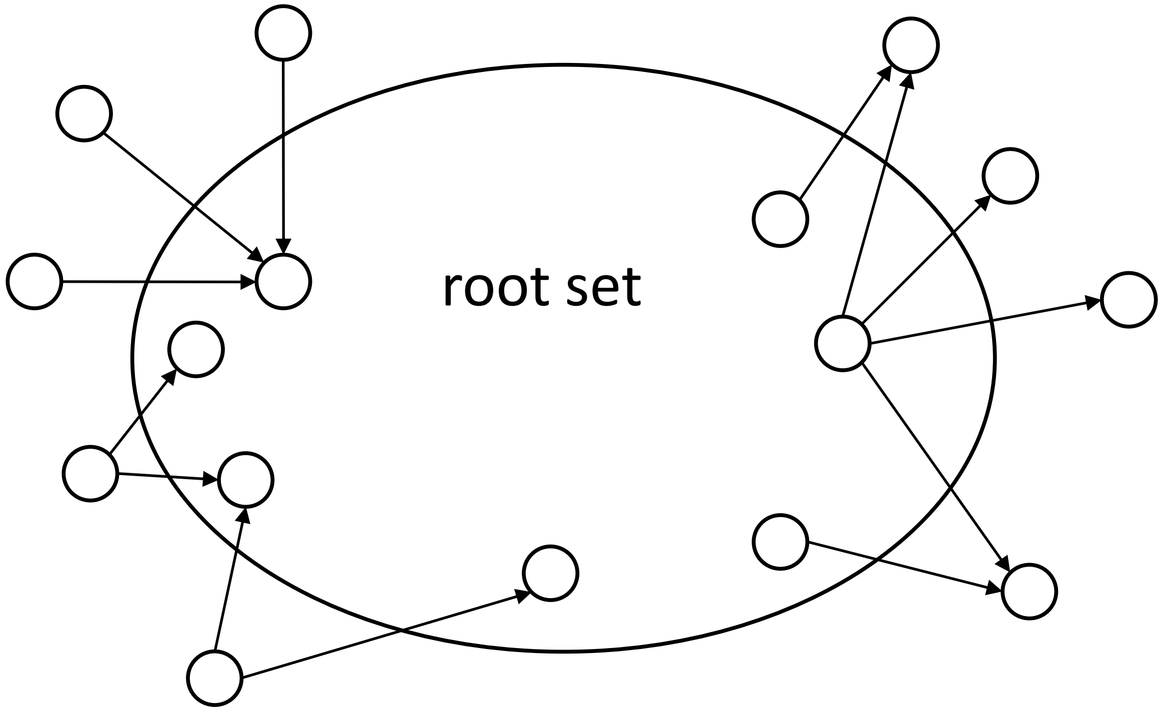
The root set

Root set and base set (1)



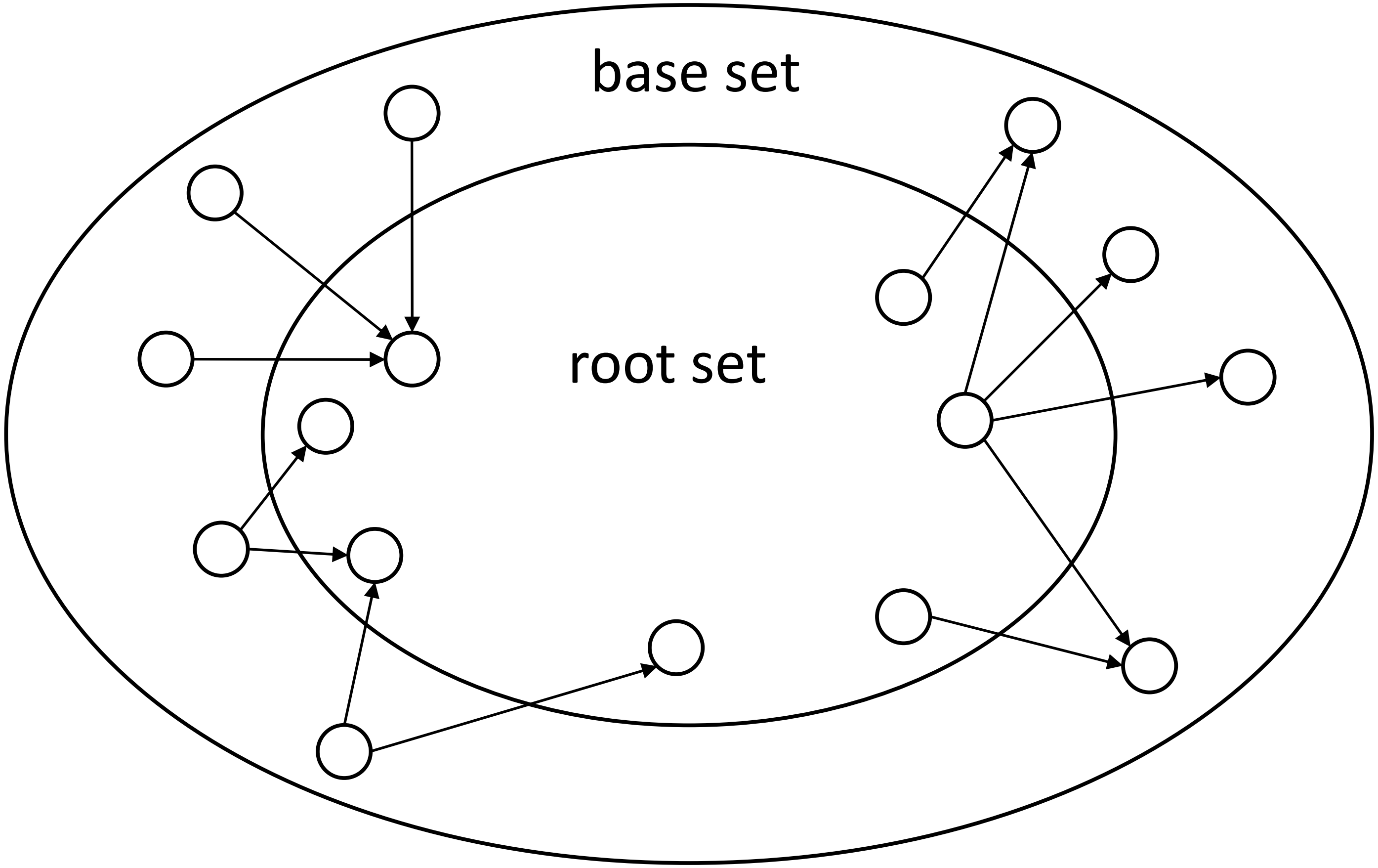
Nodes that root set nodes link to

Root set and base set (1)



Nodes that link to root set nodes

Root set and base set (1)



The base set

Root set and base set (2)

Root set and base set (2)

- Root set typically has 200-1000 nodes.

Root set and base set (2)

- Root set typically has 200-1000 nodes.
- Base set may have up to 5000 nodes.

Root set and base set (2)

- Root set typically has 200-1000 nodes.
- Base set may have up to 5000 nodes.
- Computation of base set, as shown on previous slide:

Root set and base set (2)

- Root set typically has 200-1000 nodes.
- Base set may have up to 5000 nodes.
- Computation of base set, as shown on previous slide:
  - Follow outlinks by parsing the pages in the root set

Root set and base set (2)

- Root set typically has 200-1000 nodes.
- Base set may have up to 5000 nodes.
- Computation of base set, as shown on previous slide:
  - Follow outlinks by parsing the pages in the root set
  - Find *d*'s inlinks by searching for all pages containing a link to *d*

Hub and authority scores

Hub and authority scores

- Compute for each page *d* in the base set a hub score *h*(*d*) and an authority score *a*(*d*)

Hub and authority scores

- Compute for each page *d* in the base set a hub score *h*(*d*) and an authority score *a*(*d*)
- Initialization: for all *d*: *h*(*d*) = 1, *a*(*d*) = 1

Hub and authority scores

- Compute for each page d in the base set a hub score $h(d)$ and an authority score $a(d)$
- Initialization: for all d: $h(d) = 1$, $a(d) = 1$
- Iteratively update all $h(d)$, $a(d)$

Hub and authority scores

- Compute for each page d in the base set a hub score $h(d)$ and an authority score $a(d)$
- Initialization: for all d: $h(d) = 1$, $a(d) = 1$
- Iteratively update all $h(d)$, $a(d)$
- After convergence:

Hub and authority scores

- Compute for each page *d* in the base set a hub score *h*(*d*) and an authority score *a*(*d*)
- Initialization: for all *d*: *h*(*d*) = 1, *a*(*d*) = 1
- Iteratively update all *h*(*d*), *a*(*d*)
- After convergence:
  - Output pages with highest *h* scores as top hubs

Hub and authority scores

- Compute for each page *d* in the base set a hub score *h*(*d*) and an authority score *a*(*d*)
- Initialization: for all *d*: *h*(*d*) = 1, *a*(*d*) = 1
- Iteratively update all *h*(*d*), *a*(*d*)
- After convergence:
  - Output pages with highest *h* scores as top hubs
  - Output pages with highest *a* scores as top authrities

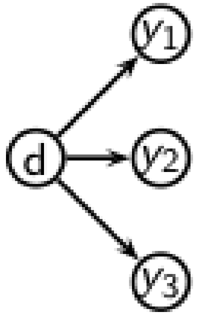
Hub and authority scores

- Compute for each page *d* in the base set a hub score *h*(*d*) and an authority score *a*(*d*)
- Initialization: for all *d*: *h*(*d*) = 1, *a*(*d*) = 1
- Iteratively update all *h*(*d*), *a*(*d*)
- After convergence:
  - Output pages with highest *h* scores as top hubs
  - Output pages with highest *a* scores as top authrities
  - So we output two ranked lists

Iterative update

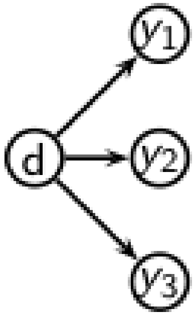
Iterative update

- For all *d*: *h*(*d*) = $\sum_{d \rightarrow y} a(y)$

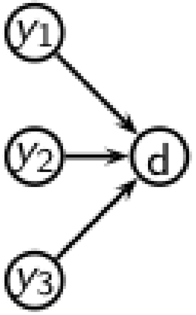


Iterative update

- For all *d*: *h*(*d*) = $\sum_{d \to y} a(y)$

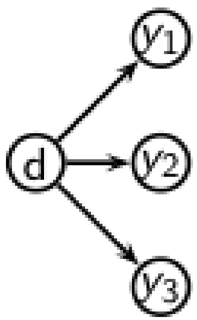


- For all *d*: *a*(*d*) = $\sum_{y \to d} h(y)$

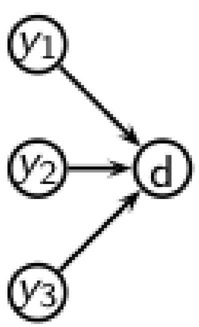


Iterative update

- For all *d*: $h(d) = \sum_{d \to y} a(y)$



- For all *d*: $a(d) = \sum_{y \to d} h(y)$



- Iterate these two steps until convergence

Details

Details

- Scaling

Details

- Scaling
  - To prevent the *a*() and *h*() values from getting too big, can scale down after each iteration.

Details

- Scaling
  - To prevent the *a*() and *h*() values from getting too big, can scale down after each iteration.
  - Scaling factor doesn't really matter.

Details

- Scaling
  - To prevent the *a*() and *h*() values from getting too big, can scale down after each iteration.
  - Scaling factor doesn't really matter.
  - We care about the relative (as opposed to absolute) values of the scores.

Details

- Scaling
  - To prevent the *a*() and *h*() values from getting too big, can scale down after each iteration.
  - Scaling factor doesn't really matter.
  - We care about the relative (as opposed to absolute) values of the scores.
- In most cases, the algorithm converges after a few iterations.

Authorities for query [Chicago Bulls]

Authorities for query [Chicago Bulls]

0.85	www.nba.com/bulls
0.25	www.essex1.com/people/jmiller/bulls.htm “da Bulls”
0.20	www.nando.net/SportServer/basketball/nba/chi.html “The Chicago Bulls”
0.15	Users.aol.com/rynocub/bulls.htm “The Chicago Bulls Home Page ”
0.13	www.geocities.com/Colosseum/6095 “Chicago Bulls”

(Ben Shaul et al, WWW8)

The authority page for [Chicago Bulls]

The authority page for [Chicago Bulls]

Hubs for query [Chicago Bulls]

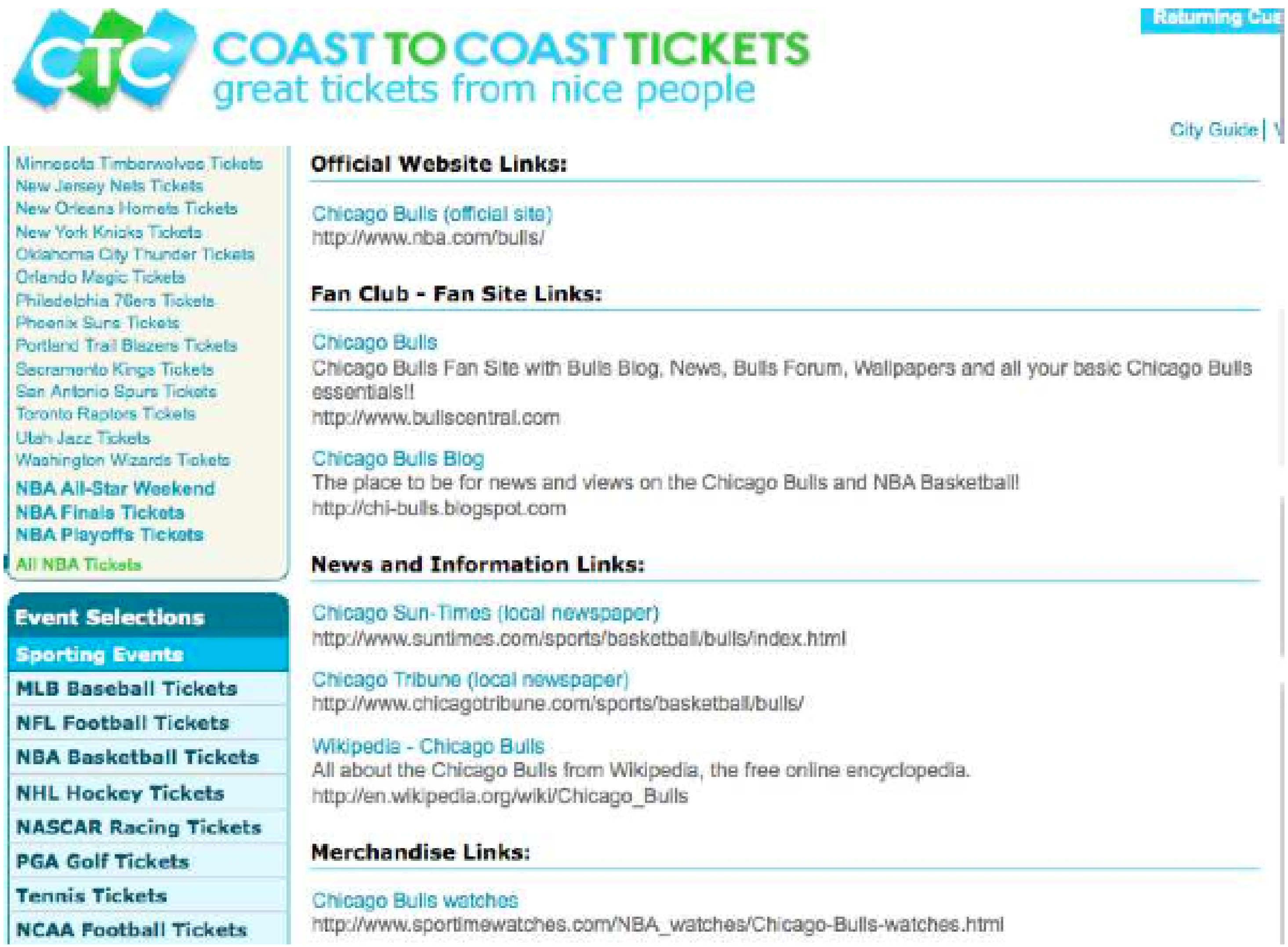
Hubs for query [Chicago Bulls]

1.62	www.geocities.com/Colosseum/1778 “Unbelieveabulls!!!!!”
1.24	www.webring.org/cgi-bin/webring?ring=chbulls “Chicago Bulls”
0.74	www.geocities.com/Hollywood/Lot/3330/Bulls.html “Chicago Bulls”
0.52	www.nobull.net/web_position/kw-search-15-M2.html “Excite Search Results: bulls ”
0.52	www.halcyon.com/wordltd/bball/bulls.html “Chicago Bulls Links”

(Ben Shaul et al, WWW8)

A hub page for [Chicago Bulls]

A hub page for [Chicago Bulls]



Hub & Authorities: Comments

Hub & Authorities: Comments

- HITS can pull together good pages regardless of page content.

Hub & Authorities: Comments

- HITS can pull together good pages regardless of page content.
- Once the base set is assembles, we only do link analysis, no text matching.

Hub & Authorities: Comments

- HITS can pull together good pages regardless of page content.
- Once the base set is assembles, we only do link analysis, no text matching.
- Pages in the base set often do not contain any of the query words.

Hub & Authorities: Comments

- HITS can pull together good pages regardless of page content.
- Once the base set is assembles, we only do link analysis, no text matching.
- Pages in the base set often do not contain any of the query words.
- In theory, an English query can retrieve Japanese-language pages!

Hub & Authorities: Comments

- HITS can pull together good pages regardless of page content.
- Once the base set is assembles, we only do link analysis, no text matching.
- Pages in the base set often do not contain any of the query words.
- In theory, an English query can retrieve Japanese-language pages!
  - If supported by the link structures between English and Japanese pages!

Hub & Authorities: Comments

- HITS can pull together good pages regardless of page content.
- Once the base set is assembles, we only do link analysis, no text matching.
- Pages in the base set often do not contain any of the query words.
- In theory, an English query can retrieve Japanese-language pages!
  - If supported by the link structures between English and Japanese pages!
- Danger: topic drift – the pages found by following links may not be related to the original query.

Proof of convergence

Proof of convergence

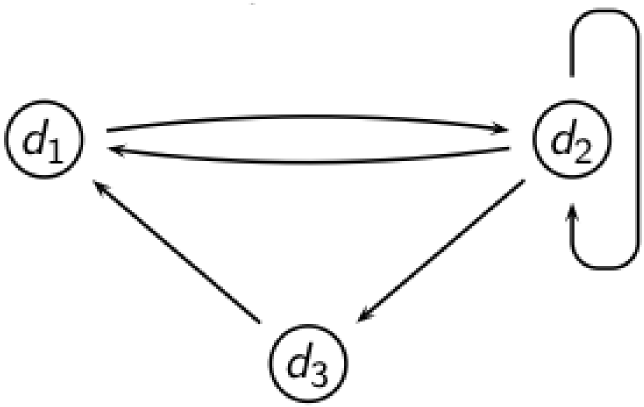
- We define an $N \times N$ adjacency matrix A. (We called this the link matrix earlier).

Proof of convergence

- We define an $N \times N$ adjacency matrix A. (We called this the link matrix earlier).
- For $1 \leq i, j \leq N$, the matrix entry A_{ij} tells us whether there is a link from page i to page j ($A_{ij} = 1$) or not ($A_{ij} = 0$).

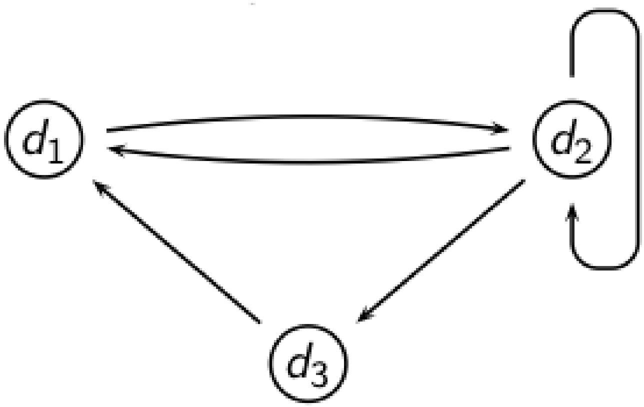
Proof of convergence

- We define an $N \times N$ adjacency matrix A. (We called this the link matrix earlier).
- For $1 \leq i, j \leq N$, the matrix entry A_{ij} tells us whether there is a link from page i to page j (A_{ij} = 1) or not (A_{ij} = 0).
- Example:



Proof of convergence

- We define an $N \times N$ adjacency matrix A. (We called this the link matrix earlier).
- For $1 \leq i, j \leq N$, the matrix entry A_{ij} tells us whether there is a link from page i to page j ($A_{ij} = 1$) or not ($A_{ij} = 0$).
- Example:



	d_1	d_2	d_3
d_1	0	1	0
d_2	1	1	1
d_3	1	0	0

Write update rules as matrix operations

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1,\ldots,}h_N)$ as the vector of hub scores. h is the hub score of page d .

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1, \ldots ,} h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1, \ldots ,} h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \to y} a(y)$ as a matrix operation: $\vec{h} = A\vec{a} \ldots$

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1,...,}h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \to y} a(y)$ as a matrix operation: $\vec{h} = A\vec{a}$ . . .
- . . . and we can write $a(d) = \sum_{y \to d} h(y)$ as a $a = A^T h$

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1, \ldots,} h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \to y} a(y)$ as a matrix operation:
  $\vec{h} = A\vec{a} \ldots$
- . . . and we can write $a(d) = \sum_{y \to d} h(y)$ as a $a = A^T h$
- HITS algorithm in matrix notation:

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1,...,}h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \to y} a(y)$ as a matrix operation: $\vec{h} = A\vec{a}$ . . .
- . . . and we can write $a(d) = \sum_{y \to d} h(y)$ as a $a = A^T h$
- HITS algorithm in matrix notation:
  - Compute $\vec{h} = A\vec{a}$

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1,\ldots,}h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \to y} a(y)$ as a matrix operation: $\vec{h} = A\vec{a}$ . . .
- . . . and we can write $a(d) = \sum_{y \to d} h(y)$ as a $a = A^T h$
- HITS algorithm in matrix notation:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T \vec{h}$

Write update rules as matrix operations

- Define the hub vector $\vec{h} = (h_{1,\ldots,}h_N)$ as the vector of hub scores. h is the hub score of page d .
- Similarity for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \to y} a(y)$ as a matrix operation: $\vec{h} = A\vec{a}$ . . .
- . . . and we can write $a(d) = \sum_{y \to d} h(y)$ as a $a = A^T h$
- HITS algorithm in matrix notation:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T \vec{h}$
  - Iterate until convergence

HITS as eigenvector problem

- HITS algorithm in matrix notation. Iterate:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T\vec{h}$

HITS as eigenvector problem

- HITS algorithm in matrix notation. Iterate:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T\vec{h}$
- By substitution we get: $\vec{h} = AA^T\vec{h}$ and $\vec{a} = A^TA\vec{a}$

HITS as eigenvector problem

- HITS algorithm in matrix notation. Iterate:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T\vec{h}$
- By substitution we get: $\vec{h} = AA^T\vec{h}$ and $\vec{a} = A^TA\vec{a}$
- Thus, $\vec{h}$ is an eigenvector of AA^T and $\vec{a}$ is an eigenvector of A^TA.

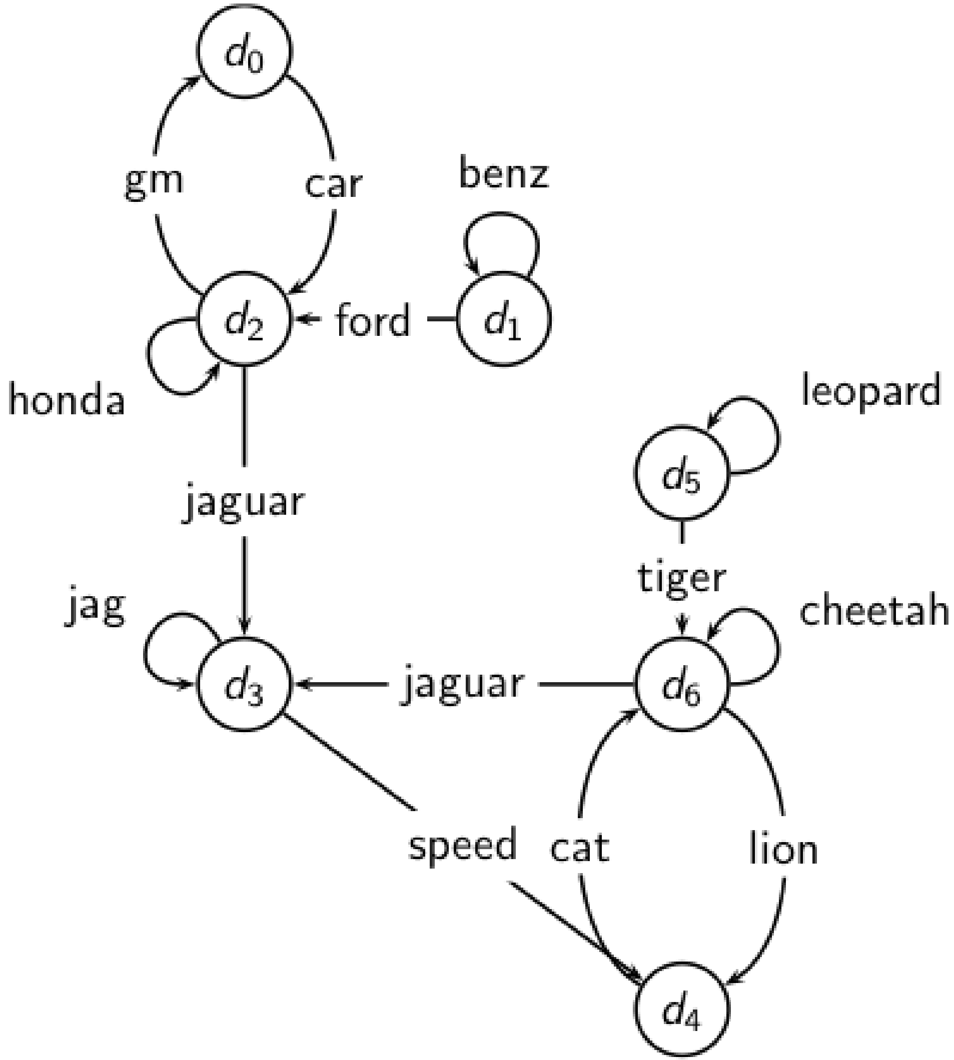
HITS as eigenvector problem

- HITS algorithm in matrix notation. Iterate:
    - Compute $\vec{h} = A\vec{a}$
    - Compute $\vec{a} = A^T\vec{h}$
- By substitution we get: $\vec{h} = AA^T\vec{h}$ and $\vec{a} = A^TA\vec{a}$
- Thus, $\vec{h}$ is an eigenvector of AA^T and $\vec{a}$ is an eigenvector of A^TA.
- So the HITS algorithm is actually a special case of the power merthod and hub and authority scores are eigenvector values.

HITS as eigenvector problem

- HITS algorithm in matrix notation. Iterate:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T\vec{h}$
- By substitution we get: $\vec{h} = AA^T\vec{h}$ and $\vec{a} = A^TA\vec{a}$
- Thus, $\vec{h}$ is an eigenvector of AA^T and $\vec{a}$ is an eigenvector of A^TA.
- So the HITS algorithm is actually a special case of the power merthod and hub and authority scores are eigenvector values.
- HITS and PageRank both formalize link analysis as eigenvector problems.

Example web graph



Raw matrix A for HITS

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	2	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	2	1	0	1

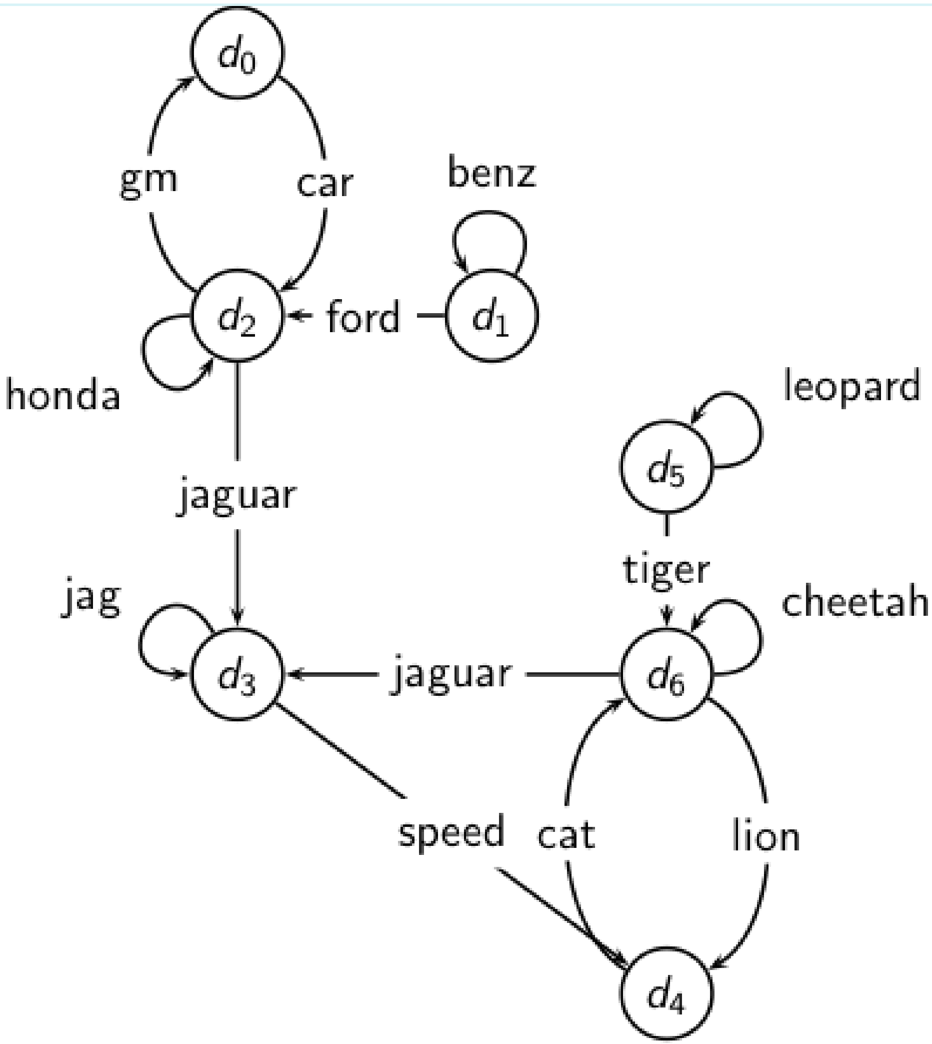
Hub vectors h_0, $\vec{h}_i = \frac{1}{d_i} A^* a_i,\ i \geq 1$

	$\vec{h}_0$	$\vec{h}_1$	$\vec{h}_2$	$\vec{h}_3$	$\vec{h}_4$	$\vec{h}_5$
d_0	0.14	0.06	0.04	0.04	0.03	0.03
d_1	0.14	0.08	0.05	0.04	0.04	0.04
d_2	0.14	0.28	0.32	0.33	0.33	0.33
d_3	0.14	0.14	0.17	0.18	0.18	0.18
d_4	0.14	0.06	0.04	0.04	0.04	0.04
d_5	0.14	0.08	0.05	0.04	0.04	0.04
d_6	0.14	0.30	0.33	0.34	0.35	0.35

Authority vector $\vec{a} = \frac{1}{c_i} A^T * \vec{h}_{i-1}, i \geq 1$

	a_1	$\vec{a}_2$	$\vec{a}_3$	$\vec{a}_4$	$\vec{a}_5$	$\vec{a}_6$	$\vec{a}_7$
d_0	0.06	0.09	0.10	0.10	0.10	0.10	0.10
d_1	0.06	0.03	0.01	0.01	0.01	0.01	0.01
d_2	0.19	0.14	0.13	0.12	0.12	0.12	0.12
d_3	0.31	0.43	0.46	0.46	0.46	0.47	0.47
d_4	0.13	0.14	0.16	0.16	0.16	0.16	0.16
d_5	0.06	0.03	0.02	0.01	0.01	0.01	0.01
d_6	0.19	0.14	0.13	0.13	0.13	0.13	0.13

Example web graph



	a	h
d_0	0.10	0.03
d_1	0.01	0.04
d_2	0.12	0.33
d_3	0.47	0.18
d_4	0.16	0.04
d_5	0.01	0.04
d_6	0.13	0.35

Top-ranked pages

Top-ranked pages

- Pages with highest in-degree: d_2, d_3, d_6

Top-ranked pages

- Pages with highest in-degree: $d_{2,}$ $d_{3,}$ d_6
- Pages with highest out-degree: $d_{2,}$ d_6

Top-ranked pages

- Pages with highest in-degree: $d_{2,}$ $d_{3,}$ d_6
- Pages with highest out-degree: $d_{2,}$ d_6
- Pages with highest PageRank: d_6

Top-ranked pages

- Pages with highest in-degree: $d_{2,}$ $d_{3,}$ d_6
- Pages with highest out-degree: $d_{2,}$ d_6
- Pages with highest PageRank: d_6
- Pages with highest in-degree: d_6 (close: d_2)

Top-ranked pages

- Pages with highest in-degree: d_2, d_3, d_6
- Pages with highest out-degree: d_2, d_6
- Pages with highest PageRank: d_6
- Pages with highest in-degree: d_6 (close: d_2)
- Pages with highest authority score: d_3

PageRank vs. HITS: Discussion

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to.

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to.
- These two are orthogonal.

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to.
- These two are orthogonal.
  - We could also apply HITS to the entire web and PageRank to a small base set.

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to.
- These two are orthogonal.
  - We could also apply HITS to the entire web and PageRank to a small base set.
- Claim: On the web, a good hub almost always is also a good authority.

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to.
- These two are orthogonal.
  - We could also apply HITS to the entire web and PageRank to a small base set.
- Claim: On the web, a good hub almost always is also a good authority.
- The actual difference between PageRank ranking and HITS ranking is therefore not as large as one might expect.

Exercise

Exercise

- Why is a good hub almost always also a good authority?

Take-away today

- Anchor text: What exactly are links on the web and why are they important for IR?
- Citation analysis: the mathematical foundation of PageRank and link-based ranking
- PageRank: the original algorithm that was used for link-based ranking on the web
- Hubs & Authorities: an alternative link-based ranking algorithm

Resources

- Chapter 21 of IIR
- Resources at `http://ifnlp.org/ir`
  - American Mathematical Society article on PageRank (popular science style)
  - Jon Kleinberg's home page (main person behind HITS)
  - A Google bomb and its defusing
  - Google's official description of PageRank: *PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results.*