MU5IN075
Network Analysis and Mining
11. Search Engine Algorithms

Esteban Bautista-Ruiz, Lionel Tabourier

LIP6 – CNRS and Sorbonne Université

first_name.last_name@lip6.fr

November 30, 2021

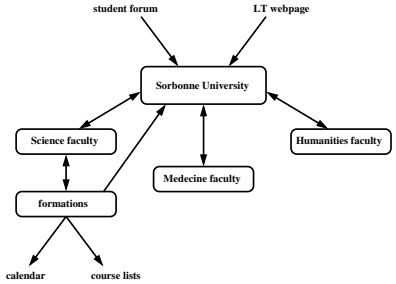
Outline

Graph of the World Wide Web

Mostly adapted from Jon Kleinberg's edX course - *Networks, Crowds and Markets*

Web graph: nodes = page / link: hyperlink $\Rightarrow$ **directed**

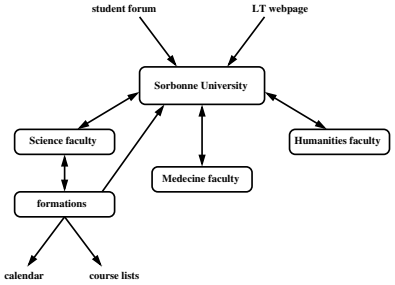
The web: large graph, strong asymmetry (*ex: SU website*)



$\longrightarrow$ notions of direction, **flow** of information

A large-scale map of the web?

Around 40 billion pages (`WorldWideWebSize.com`)
$\Rightarrow$ comprehensive map would be unreadable
Try to find a **large-scale scheme**



$\Rightarrow$ **strongly connected component**

A large-scale map of the web?

Around 40 billion pages (`WorldWideWebSize.com`)
⇒ comprehensive map would be unreadable
Try to find a **large-scale scheme**

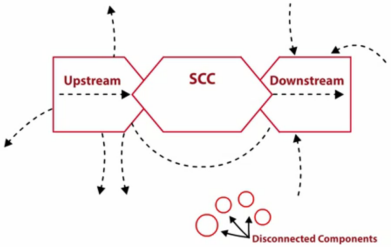
Definition

- **directed connectedness**: can go from any *i* to any *j*
- **maximality**: cannot add any node

The bow-tie map

Giant connected component
Several large connected components would be very unstable...

The bow-tie:

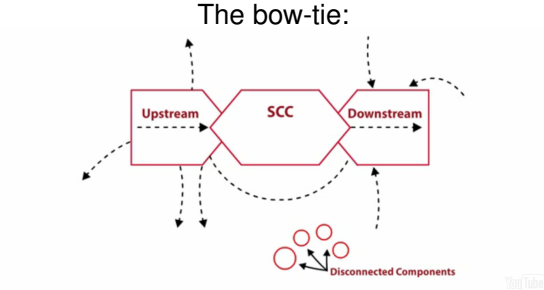


Question to answer: we have a global map...
but how to locate precise information?

The bow-tie map

Giant connected component
Several large connected components would be very unstable...

The bow-tie:



Question to answer: we have a global map...
but how to locate precise information?

Outline

Information retrieval

Historically

Field of how to locate an information in a data repository dates back to 50's and 60's for highly specialized experts:

- librarians *(fr: bibliothécaire)*
- patent attorneys *(fr: juriste conseil en brevets)*

Traditional principle

1. individual: information need (book, patent, law, . . .)
2. request to an expert (librarian, attorney, . . .)
3. expert select a subfield
4. then targets and orders relevant results

3 and 4 are the information retrieval phases

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Information retrieval

Traditional principle

1. individual: information need (book, patent, law, . . .)
2. request to an expert (librarian, attorney, . . .)
3. expert select a subfield
4. then targets and orders relevant results

3 and 4 are the information retrieval phases

What is different on the web?

- scale (billion pages)
- retriever (amateur instead of professional)
- dynamical aspect (content changes fast)

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Information retrieval on the web

What is different on the web?

- scale (billion pages)
- retriever (amateur instead of professional)
- dynamical aspect (content changes fast)

$\Rightarrow$ need for automatic information retrieval

Selection of a subfield

Natural Language Processing $\rightarrow$ out of the scope of this course

Targeting and ordering results

What we are about to discuss: let's the web vote

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Information retrieval on the web

What is different on the web?

- scale (billion pages)
- retriever (amateur instead of professional)
- dynamical aspect (content changes fast)

$\Rightarrow$ need for automatic information retrieval

Selection of a subfield

Natural Language Processing $\rightarrow$ out of the scope of this course

Targeting and ordering results

What we are about to discuss: let's the web vote

Information retrieval on the web

What is different on the web?

- scale (billion pages)
- retriever (amateur instead of professional)
- dynamical aspect (content changes fast)

$\Rightarrow$ need for automatic information retrieval

Selection of a subfield

Natural Language Processing $\rightarrow$ out of the scope of this course

Targeting and ordering results

What we are about to discuss: let's the web vote

Endorsement: hubs and authorities

fr: appui, approbation

Who to trust?

hyperlink induced topic-search

Goal: identify who is a good hub, who is a good authority

- **hub**: good at referencing what you look for
- **authority**: what you look for

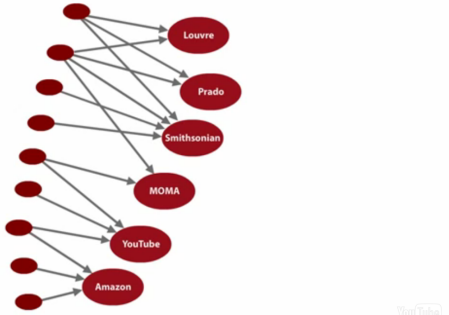
warning: hubs $\neq$ what we have defined in the first courses

hubs endorse authorities

HITS algorithm: qualitative description

Authoritative sources in a hyperlinked environment - *Kleinberg, 1999*

Aim at ranking relevant pages (ex: *"museum"*)

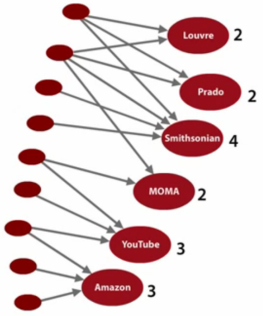


initial ranking: take a large sample of the web, then vote

HITS algorithm: qualitative description

Authoritative sources in a hyperlinked environment - *Kleinberg, 1999*

Aim at ranking relevant pages (ex: *"museum"*)

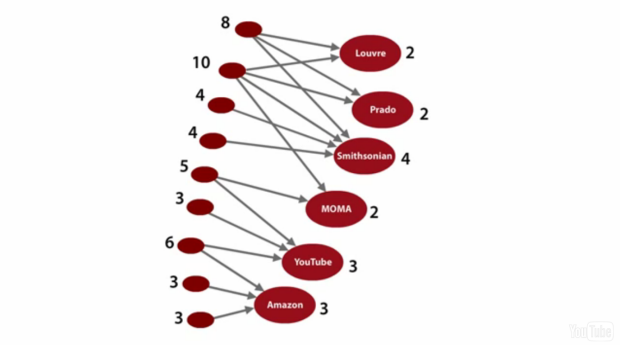


vote: number of ingoing links to supposed authorities...

HITS algorithm: qualitative description

Authoritative sources in a hyperlinked environment - *Kleinberg, 1999*

Aim at ranking relevant pages (ex: *"museum"*)

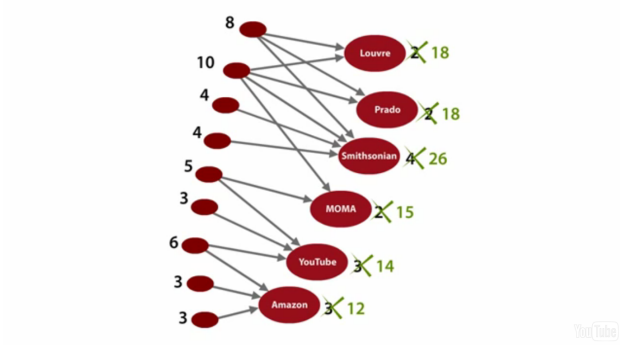


vote: ...then weight of outgoing links from supposed hubs

HITS algorithm: qualitative description

Authoritative sources in a hyperlinked environment - *Kleinberg, 1999*

Aim at ranking relevant pages (ex: *"museum"*)

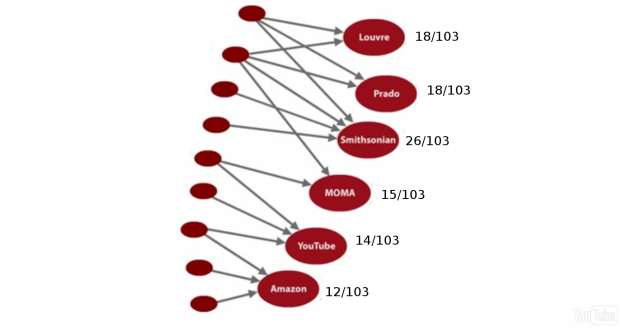


revote $\Rightarrow$ giving more weight to reliable sources

HITS algorithm: qualitative description

Authoritative sources in a hyperlinked environment - *Kleinberg, 1999*

Aim at ranking relevant pages (ex: *"museum"*)



normalization and **iteration**

HITS algorithm: summary

Authoritative sources in a hyperlinked environment - *Kleinberg, 1999*

More formally:

- **authority update:** sum hub scores pointing at authority
- **hub update:** sum authority scores pointed by hub
- **normalization** and **iteration**

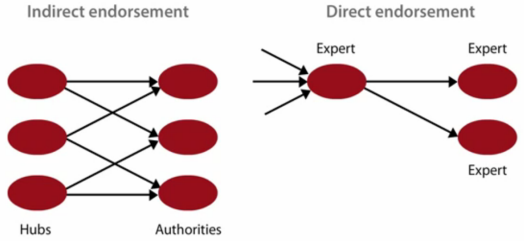
Convergence to a unique stable state (admitted)

Outline

Direct endorsement on the web

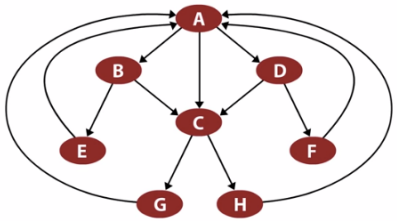
HITS algorithm:
good hubs, good authorities

Often on the web, **both roles**: *"experts"*
(examples: videos, institutions, bloggers...)



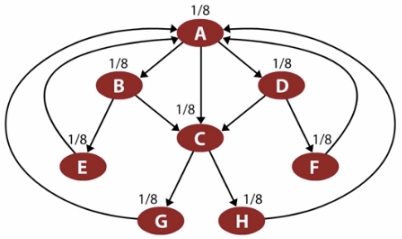
PageRank algorithm: basic principle

The PageRank citation ranking: bringing order to the Web - *Page et al., 1999*



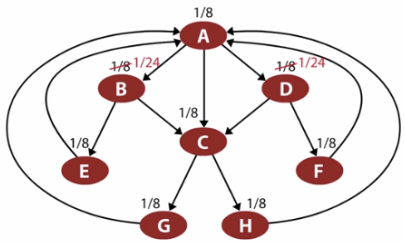
PageRank:
fluid stream through a strongly connected component:
accumulates on important pages

PageRank algorithm: basic principle



Suppose we give equal weight to all nodes in the graph,
normalized : 1/N

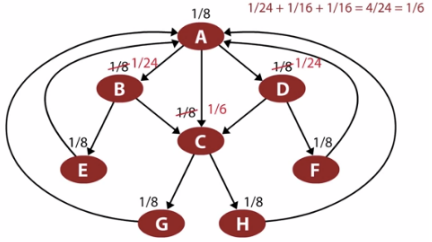
PageRank algorithm: basic principle



Update rule: simultaneously split your PR evenly to all your destinations

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

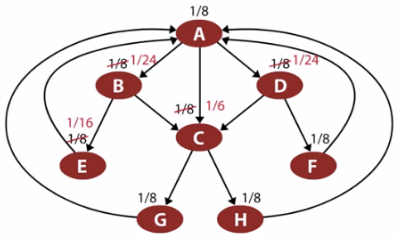
PageRank algorithm: basic principle



Update rule: simultaneously split your PR evenly to all your destinations

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

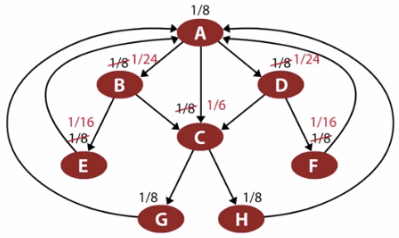
PageRank algorithm: basic principle



Update rule: simultaneously split your PR evenly to all your destinations

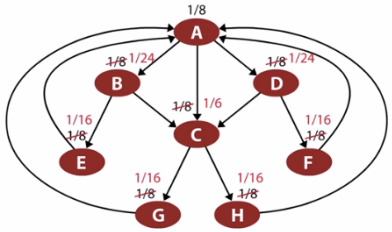
Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: basic principle



Update rule: simultaneously split your PR evenly to all your destinations

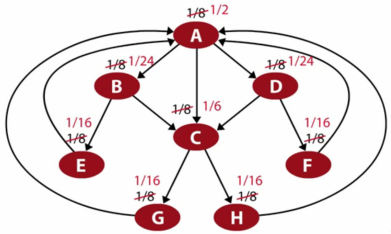
PageRank algorithm: basic principle



Update rule: simultaneously split your PR evenly to all your destinations

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: basic principle

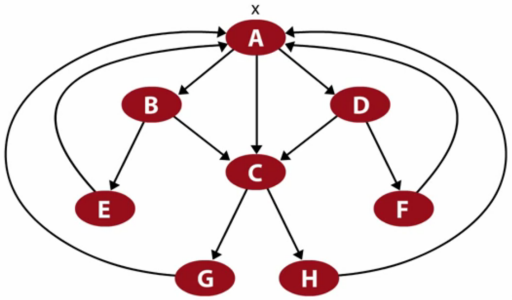


Update rule: simultaneously split your PR evenly to all your destinations

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: analytical computation

What is the final state?
admitted: only one equilibrium state

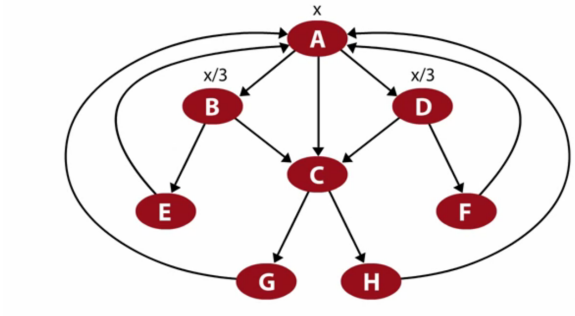


In practice: numerical updates (not analytical computations)

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: analytical computation

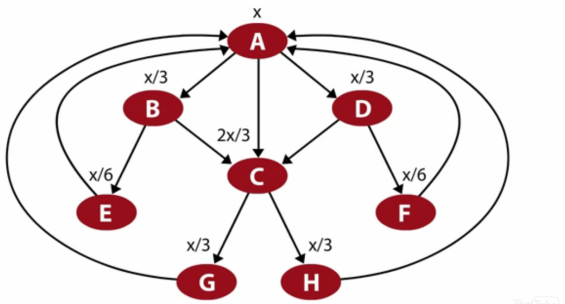
What is the final state?
admitted: only one equilibrium state



In practice: numerical updates (not analytical computations)

PageRank algorithm: analytical computation

What is the final state?
admitted: only one equilibrium state



In practice: numerical updates (not analytical computations)

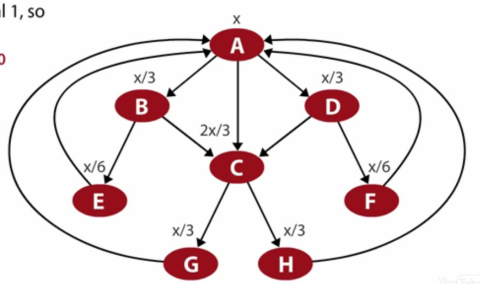
PageRank algorithm: analytical computation

What is the final state?
admitted: only one equilibrium state

$x + x/3 + x/3 + 2x/3 + x/6 + x/6 + x/3 + x/3 = 10x/3$

This must equal 1, so

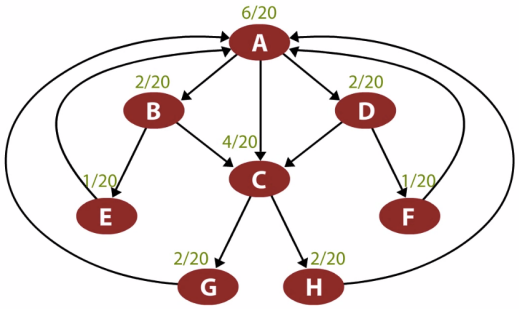
- $10x/3 = 1$
- $x = 3/10 = 6/20$



In practice: numerical updates (not analytical computations)

PageRank algorithm: analytical computation

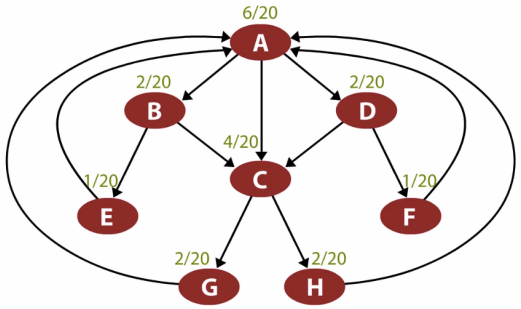
What is the final state?
admitted: only one equilibrium state



In practice: numerical updates (not analytical computations)

PageRank algorithm: analytical computation

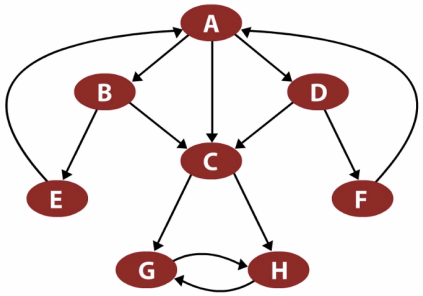
What is the final state?
admitted: only one equilibrium state



In practice: numerical updates (not analytical computations)

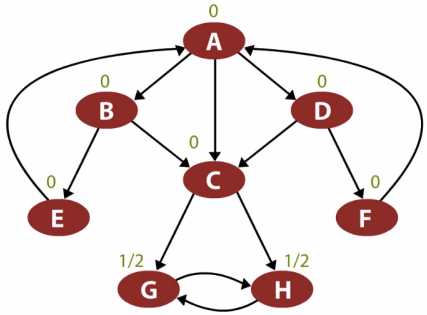
PageRank algorithm: problem with spider-traps

Suppose we modify slightly the structure:



What is the problem?

PageRank algorithm: problem with spider-traps



not a strongly connected graph...
happens a lot on the web (small company refers its own pages)
downstream of the bow-tie and no way to come back upstream
does not reflect our intuition...

PageRank algorithm: evaporation

Add evaporation:
scaling factor $s < 1$

New update rule:
$(1 - s)$ **update of pagerank**
$+ s$ **equal redistribution** of evaporation

$\Rightarrow$ giant component get most of the evaporated PR
$\Rightarrow$ find meaningful equilibrium state

PageRank algorithm: evaporation

Add evaporation:
scaling factor $s < 1$

New update rule:
$(1 - s)$ **update of pagerank**
$+ s$ **equal redistribution** of evaporation

$\Rightarrow$ giant component get most of the evaporated PR
$\Rightarrow$ find meaningful equilibrium state

PageRank algorithm: renormalization

Other problem:
at dead-ends (nodes with out-degree = 0)
PageRank disappears during the updates

add renormalization
at each step, **we rescale the PageRank to 1**

Other positive effect: avoids possible numerical drifts

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: renormalization

Other problem:
at dead-ends (nodes with out-degree = 0)
PageRank disappears during the updates

add renormalization
at each step, **we rescale the PageRank to 1**

Other positive effect: avoids possible numerical drifts

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: renormalization

Other problem:
at dead-ends (nodes with out-degree = 0)
PageRank disappears during the updates

add renormalization
at each step, **we rescale the PageRank to 1**

Other positive effect: avoids possible numerical drifts

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

PageRank algorithm: summary

The PageRank citation ranking: bringing order to the Web - *Page et al., 1999*

More formally:

- update phase:

$$x_i \leftarrow (1-s) \sum_{j \in pred(i)} x_j / d_{out}(j) + s/N$$

- renormalization phase:

$$x_i \leftarrow K.x_i \quad \text{with } K \text{ such that} \quad \sum_i x_i = 1$$

- iterate until convergence

Convergence to a unique stable state (admitted)

Analogy to a random walk with teleportation

Suppose you consider **a random walker** on the web graph:

- equal probability to take any link to a page
  $\leftrightarrow$ fluid streaming on the web
- with small probability to teleport anywhere
  $\leftrightarrow$ evaporation

PageRank = probability to be on a page in stable state

Rk: starting from whatever state, same equilibrium

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Analogy to a random walk with teleportation

Suppose you consider **a random walker** on the web graph:

- equal probability to take any link to a page
  $\leftrightarrow$ fluid streaming on the web
- with small probability to teleport anywhere
  $\leftrightarrow$ evaporation

PageRank = probability to be on a page in stable state

Rk: starting from whatever state, same equilibrium

Outline

Random Walk: Matrix-Vector Multiplication

Given a $n \times n$ matrix A and a vector V (n by 1 matrix)
compute vector U (n by 1 matrix) such that:

$$U = A \times V$$

where $\forall i \in [\![1, n]\!]$, $U_i = \sum_{j=1}^{n} A_{ij} \times V_j$.

Classic sparse matrix/vector multiplication

Algo 1: Sparse matrix/vector multiplication

function SPARSEPRODMATVECT(A, V)
  for i from 1 to n **do**
    $U[i] \leftarrow 0$
    for j from 1 to n such that $A[i][j] \neq 0$ **do**
      $U[i] \leftarrow U[i] + A[i][j] \times V[j]$
  return U

Questions: Data structure? Memory consumption? Running time?

PageRank spreading

X vector which stores the PageRank of the nodes: **state vector**
$T[i][j]$ probability to go from i to j: **transition matrix**

At each step:

- without evaporation:
  $X \leftarrow \text{SPARSEMATVECTPROD}(T, X)$
- with evaporation:
  $X \leftarrow \text{SPARSEMATVECTPROD}(T, X).(1-s) + s.I$
  where I is a vector of $1/n$ values

Power iteration method for PageRank

Algo 2: Power iteration

function POWERITERATION(T, t, s)
  $X \leftarrow$ initial state (random)
  for i from 1 to t **do**
    $X \leftarrow \text{SPARSEMATVECTPROD}(T, X).(1-s) + s.I$
    $X \leftarrow \text{NORMALIZE}(X)$ // $\forall i \in [\![1, n]\!]$, $X[i] \leftarrow \frac{X[i]}{||X||_1}$
  return X

- T is the transition matrix:
  if there is a link $(u \rightarrow v)$, $T[u][v] = 1/d_{out}(u)$
  otherwise $T[u][v] = 0$
- I is a vector of $1/n$ values

Questions: Running time? Role of normalization?

Role of normalization

- in all cases avoid numerical drift
- in case of dead-ends

  u is a dead-end: $\forall v, T[u][v] = 0 \Rightarrow$ PageRank disappears...
  in other words, $||X||_1 < 1$

  when normalizing, we re-inject PageRank to all nodes
  $X[i] \leftarrow \frac{X[i]}{||X||_1} \Rightarrow X$ is normalized again

  remark: we could also do $X[i] \leftarrow X[i] + \frac{1-||X||_1}{n}$
  which is another normalization

$\Rightarrow$ guarantee to keep a constant PageRank in the network

Outline

Offline phase

PageRank used to be Google SE backbone
but a lot of other elements to consider . . .

Indexing the web

- impossible to apply on the whole network at each query
  → indexation
- make a “local copy” of the web for offline processing
- how? *crawling* robots
  - move from page to page using hyperlinks
  - download pages on the server
  - process pages (indexing keywords)
- Rk: billion of billions of bytes ⇒ huge storage needs!

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Offline phase

PageRank used to be Google SE backbone
but a lot of other elements to consider . . .

Indexing the web

- impossible to apply on the whole network at each query
  → indexation
- make a “local copy” of the web for offline processing
- how? *crawling* robots
  - move from page to page using hyperlinks
  - download pages on the server
  - process pages (indexing keywords)
- Rk: billion of billions of bytes ⇒ huge storage needs!

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Offline phase

PageRank used to be Google SE backbone
but a lot of other elements to consider . . .

Indexing the web

- impossible to apply on the whole network at each query
  → indexation
- make a “local copy” of the web for offline processing
- how? *crawling* robots
  - move from page to page using hyperlinks
  - download pages on the server
  - process pages (indexing keywords)
- Rk: billion of billions of bytes ⇒ huge storage needs!

Offline phase

PageRank used to be Google SE backbone
but a lot of other elements to consider . . .

Indexing the web

- impossible to apply on the whole network at each query → indexation
- make a “local copy” of the web for offline processing
- how? *crawling* robots
  - move from page to page using hyperlinks
  - download pages on the server
  - process pages (indexing keywords)
- Rk: billion of billions of bytes ⇒ huge storage needs!

Research phase

Indexation is a continuous task . . .

. . . new query ⇒ research phase

Research phase

- research is achieved on the local copy
- keywords ⇒ select sub-part of the index “Web sample”

rk: even if efficient algorithms ⇒ huge computational needs!

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Research phase

Indexation is a continuous task . . .

. . . new query ⇒ research phase

Research phase

- research is achieved on the local copy
- keywords ⇒ select sub-part of the index “Web sample”

rk: even if efficient algorithms ⇒ huge computational needs!

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Research phase

Indexation is a continuous task . . .

. . . new query ⇒ research phase

Research phase

- research is achieved on the local copy
- keywords ⇒ select sub-part of the index “Web sample”

rk: even if efficient algorithms ⇒ huge computational needs!

Other mechanisms

A few other ingredients

- textual content: query matching
- history: user feedback (clicks on the result pages)
- location: many research are place related
  ex: looking for a service "bank", "pizza"
- moment in time: change order depending on events
  ex: gifts around Christmas, current events, sport game

Mix depends on the search engine
In the following: Yandex case, around 2017

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Other mechanisms

A few other ingredients

- textual content: query matching
- history: user feedback (clicks on the result pages)
- location: many research are place related
  ex: looking for a service "bank", "pizza"
- moment in time: change order depending on events
  ex: gifts around Christmas, current events, sport game

Mix depends on the search engine
In the following: Yandex case, around 2017

31/35

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Other mechanisms

A few other ingredients

- textual content: query matching
- history: user feedback (clicks on the result pages)
- location: many research are place related
  ex: looking for a service "bank", "pizza"
- moment in time: change order depending on events
  ex: gifts around Christmas, current events, sport game

Mix depends on the search engine
In the following: Yandex case, around 2017

31/35

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Other mechanisms

A few other ingredients

- textual content: query matching
- history: user feedback (clicks on the result pages)
- location: many research are place related
  ex: looking for a service "bank", "pizza"
- moment in time: change order depending on events
  ex: gifts around Christmas, current events, sport game

Mix depends on the search engine
In the following: Yandex case, around 2017

31/35

Other mechanisms

A few other ingredients

- textual content: query matching
- history: user feedback (clicks on the result pages)
- location: many research are place related
  ex: looking for a service "bank", "pizza"
- moment in time: change order depending on events
  ex: gifts around Christmas, current events, sport game

Mix depends on the search engine
In the following: Yandex case, around 2017

Find the language of a request

example: pain (fr) / pain (en)

Direct inference

If related information is available:

- geographical regions
- past research

Indirect inference

How to dismiss ambiguous statements?

Words association:

- "back + pain" → english
- "pain + boulangerie" → french

Find the language of a request

example: pain (fr) / pain (en)

Direct inference

If related information is available:

- geographical regions
- past research

Indirect inference

How to dismiss ambiguous statements?

Words association:

- "back + pain" → english
- "pain + boulangerie" → french

Find the language of a request

example: pain (fr) / pain (en)

Direct inference

If related information is available:

- geographical regions
- past research

Indirect inference

How to dismiss ambiguous statements?

Words association:

- "back + pain" → english
- "pain + boulangerie" → french

Set the extent of a query

Semantic context of a query

- unravel ambiguities
  using location, time, session history
  ex: "Casablanca" → city? restaurant? movie?
- find related words
  using synonym detection, stemming, . . .
  ex: "Charles Ier" → Charlemagne, Carolus Magnus
  ex: "advisor" → advice, advis-, . . .
- detect & correct grammar mistakes and typos
  using history over the whole population
  ex: "artic" → arctic, "algorythm" → algorithm'

Set the extent of a query

Semantic context of a query

- unravel ambiguities
  using location, time, session history
  ex: "Casablanca" → city? restaurant? movie?
- find related words
  using synonym detection, stemming, . . .
  ex: "Charles Ier" → Charlemagne, Carolus Magnus
  ex: "advisor" → advice, advis-, . . .
- detect & correct grammar mistakes and typos
  using history over the whole population
  ex: "artic" → arctic, "algorythm" → algorithm'

Set the extent of a query

Semantic context of a query

- unravel ambiguities
  using location, time, session history
  ex: "Casablanca" → city? restaurant? movie?
- find related words
  using synonym detection, stemming, . . .
  ex: "Charles Ier" → Charlemagne, Carolus Magnus
  ex: "advisor" → advice, advis-, . . .
- detect & correct grammar mistakes and typos
  using history over the whole population
  ex: "artic" → arctic, "algorythm" → algorithm'

Personalizing research

Yandex procedure about location

Two rankings per query: general vs location-based

- Crucial to decide which category
- Decision based on statistics: words co-occurrence with location-related words

What does the engine memorize?

- Language used, past queries and clicks
- Choices specific to a session
- Different time scales:
  - long scale (language, location, recurring preferences)
  - short scale (session choices)

Personalizing research

Yandex procedure about location

Two rankings per query: general vs location-based

- Crucial to decide which category
- Decision based on statistics: words co-occurrence with location-related words

What does the engine memorize?

- Language used, past queries and clicks
- Choices specific to a session
- Different time scales:
  - long scale (language, location, recurring preferences)
  - short scale (session choices)

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

Personalizing research

About statistical inference

- Essential process for data mining & machine learning
- Algorithms use past data to predict or influence future behaviors

→ to be discussed in the recommendation course

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

From the early days to a world-scale business

storage needs, computational needs, importance of data . . .
⇒ hard for a newcomer to emerge

Search engine retroaction loop

Search engines became products...

- websites creator have incentive to rank high (actually most important criterion)
- search engine designers have to adapt to rank efficiently

Various balance of ingredients

Generalists, language oriented, privacy-focused...

Introduction – Mapping the web
HITS algorithm
The PageRank algorithm: principle
The PageRank algorithm: implementation
About real-world search engines

From the early days to a world-scale business

storage needs, computational needs, importance of data . . .
⇒ hard for a newcomer to emerge

Search engine retroaction loop

Search engines became products...

- websites creator have incentive to rank high (actually most important criterion)
- search engine designers have to adapt to rank efficiently

Various balance of ingredients

Generalists, language oriented, privacy-focused...

From the early days to a world-scale business

storage needs, computational needs, importance of data . . .
$\Rightarrow$ hard for a newcomer to emerge

Search engine retroaction loop

Search engines became products...

- websites creator have incentive to rank high (actually most important criterion)
- search engine designers have to adapt to rank efficiently

Various balance of ingredients

Generalists, language oriented, privacy-focused...