Queen Mary
University of London

ECS766P Data Mining

Week 10: Web Mining

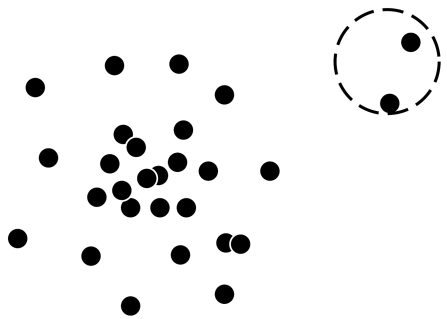
Emmanouil Benetos
emmanouil.benetos@qmul.ac.uk

November 2021

School of EECS, Queen Mary University of London

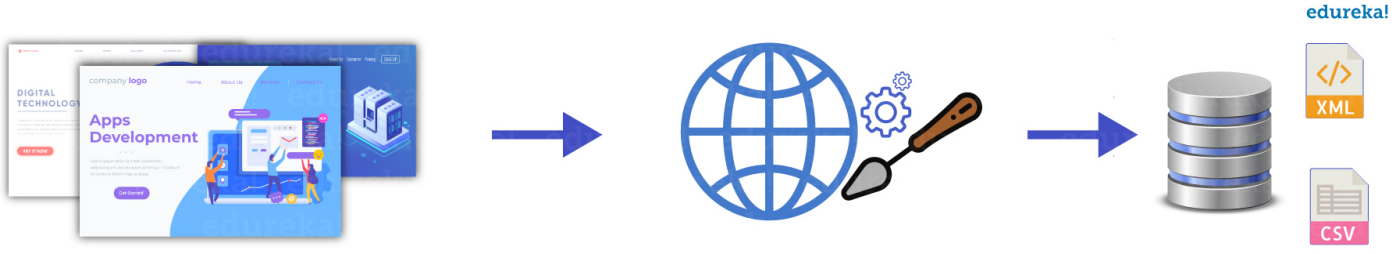
Last week: Outlier detection

- Outliers and Outlier Analysis
- Outlier Detection Methods
- Statistical Approaches
- Proximity-Based Approaches
- Clustering-Based Approaches
- Classification Approaches
- Mining Contextual and Collective Outliers



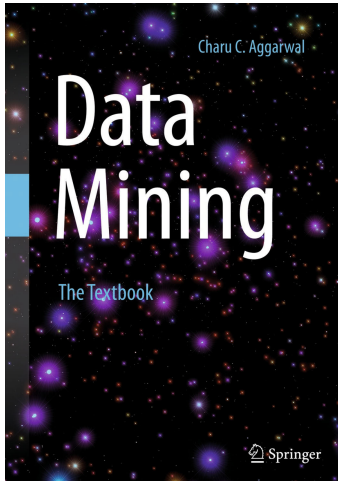
This week's contents

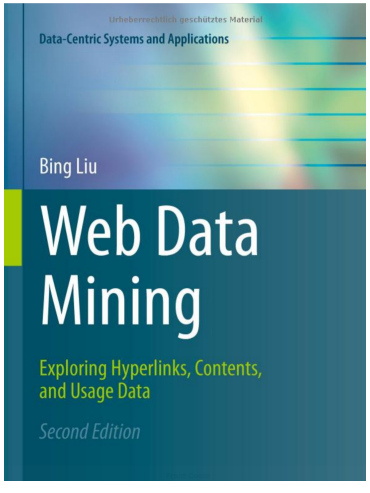
1. Six Paradigms for Today's Internet
2. Technology Review
3. Internet Mining Applications
4. Ingesting Internet data
5. Search Engine Indexing & Ranking

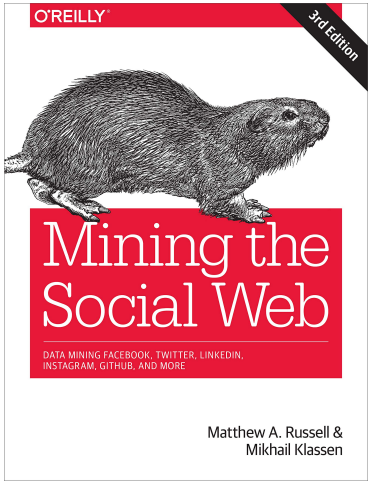


Suggested Reading

- Chapter 18 of C. C. Aggarwal, “Data Mining: The Textbook”, Springer, 2015
- B. Liu, “Web Data Mining”, 2nd Ed., Springer, 2011
- M. A. Russell and M. Klassen, “Mining the Social Web”, 3rd Ed., O’Reilly, 2019







Six Paradigms for Today's Internet

Your view

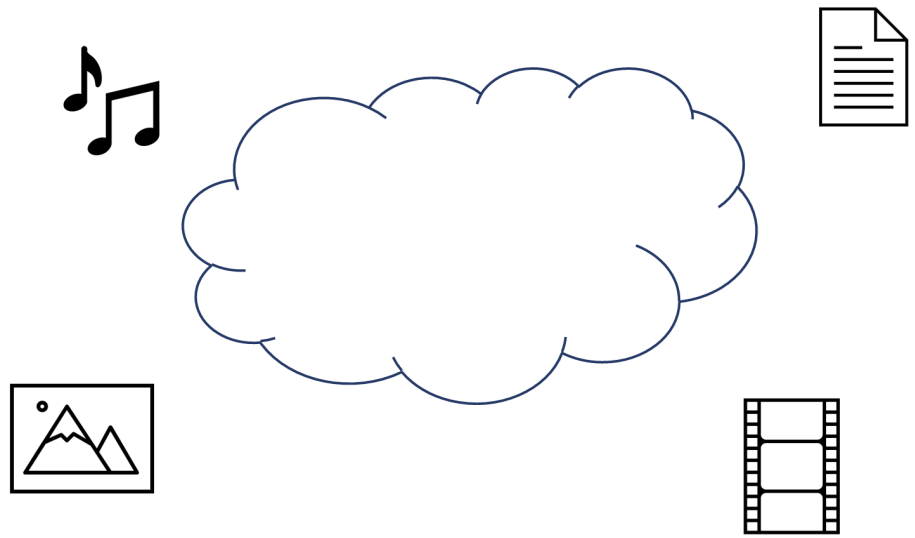
One Internet, many views

When thinking about the internet, each one of us might see a different object. What comes to your mind when you think about the Internet?

`www.menti.com`
Code:

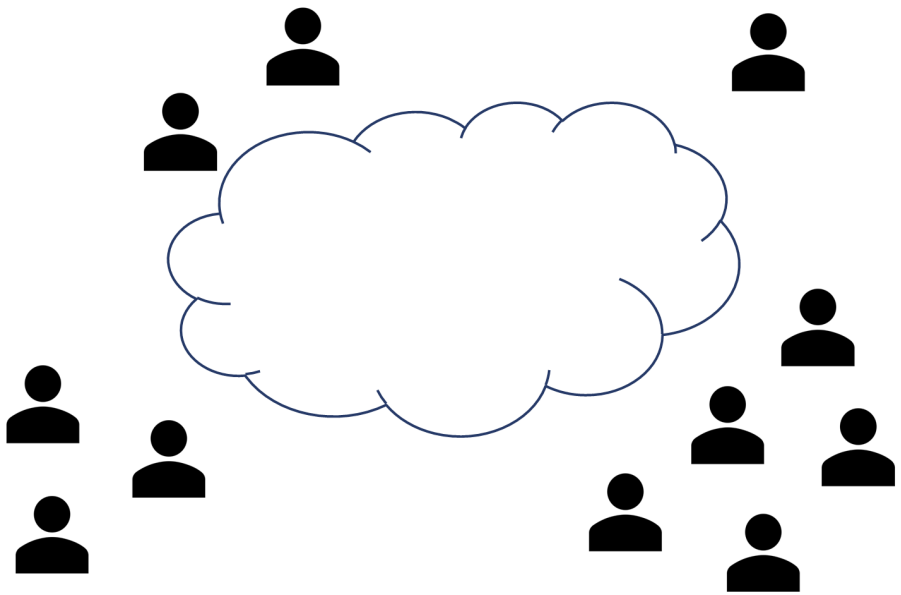
Model 1: The Library

The Internet as a repository of knowledge



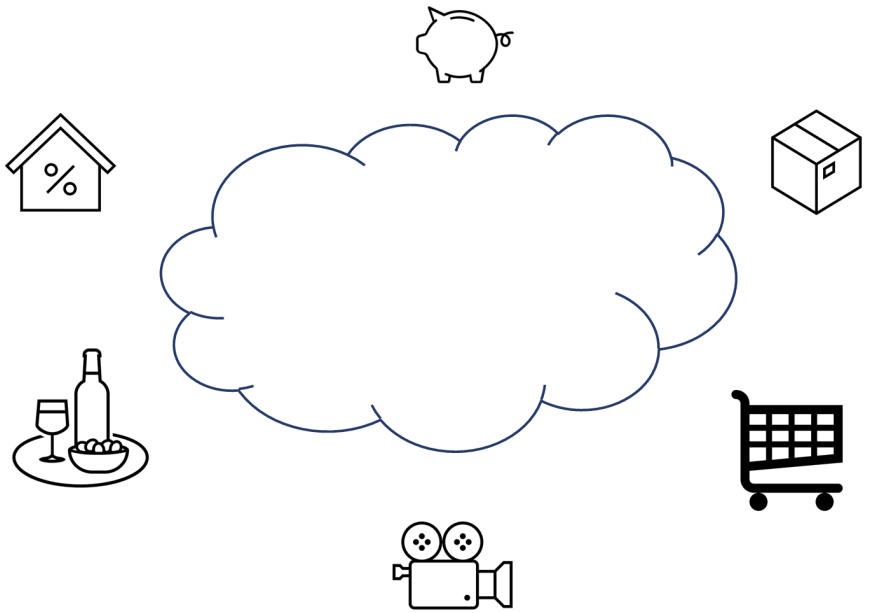
Model 2: The Cafe

The Internet as a social hub



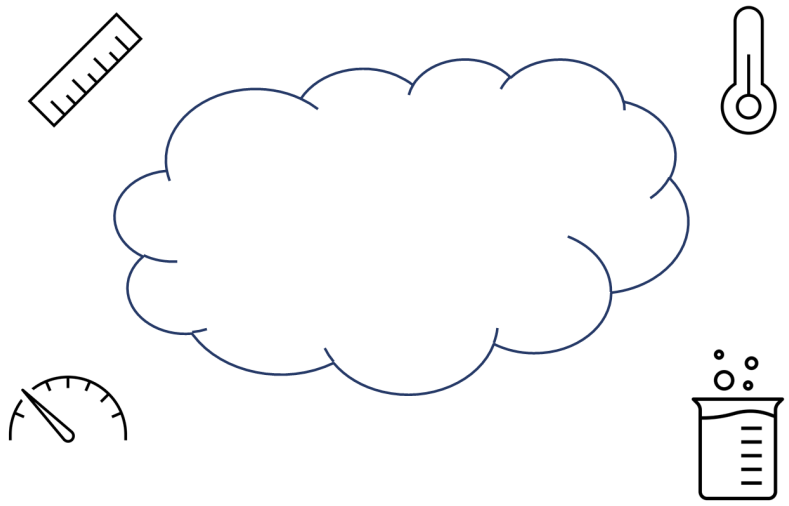
Model 3: The Mall

The Internet as a services ecosystem



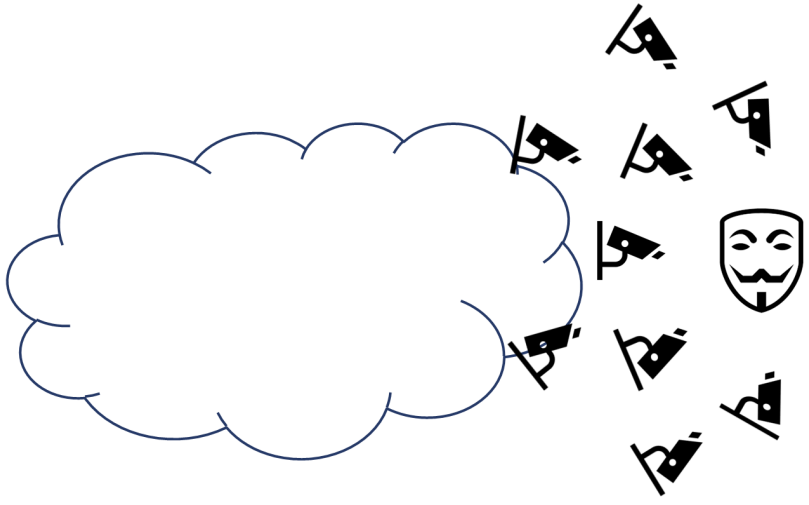
Model 4: The Lab

The Internet as sensing and experimentation infrastructure



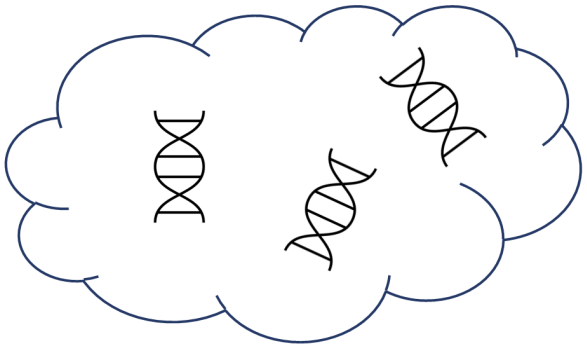
Model 5: The Big Brother

The Internet as surveillance apparatus



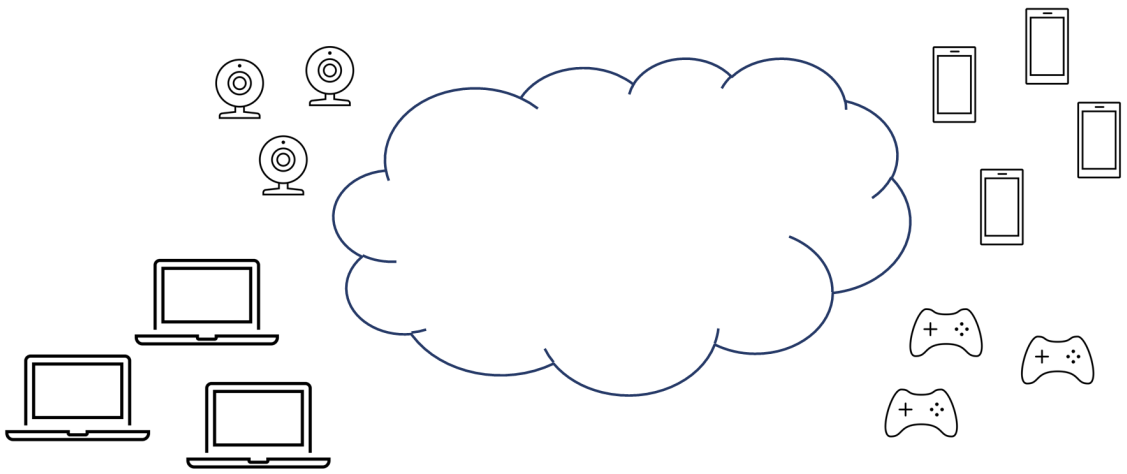
Model 6: The Organism

The Internet as a complex system



The Internet in a nutshell

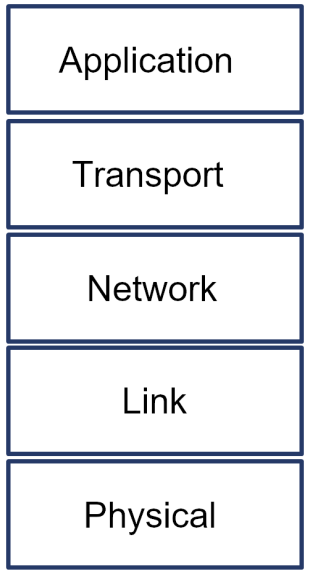
The Internet is a computer network that allows to interconnect hundreds of millions of devices worldwide, including PCs, smartphones, IoT devices, gaming consoles, servers and automobiles. As such, the Internet can be seen as the largest generator of digital data in the world.



The Internet or the Web?

The Internet is an extremely complex system. Its architecture is organised in layers serving different functions:

- Application layer: Application data is exchanged (HTTP, FTP, SMTP...).
- Transport layer: Establishes a data channel between end-devices (TCP...).
- Network layer: Routes a packet of bits (datagram) from the source end-device to the destination end-device, across multiple nodes (IP).
- Link layer: Moves a packet of bits from one node in the network to another node to which it is directly connected.
- Physical layer: Moves individual bits through a physical channel.

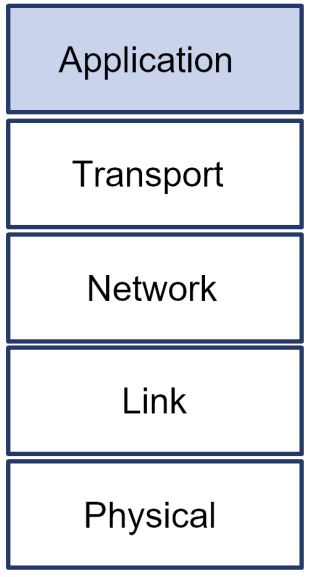


A layer provides a service to the layer above.

The Internet or the Web?

The Web is an Internet Application that uses the Hyper-Text Transfer Protocol (HTTP) to exchange data.

The Web serves as a platform for deploying many popular services, hence the early interest in mining the Web and exploiting HTTP.



Definitions

Internet Mining

Internet Mining is the application of data mining principles and strategies to analyse Internet data.

Web Mining

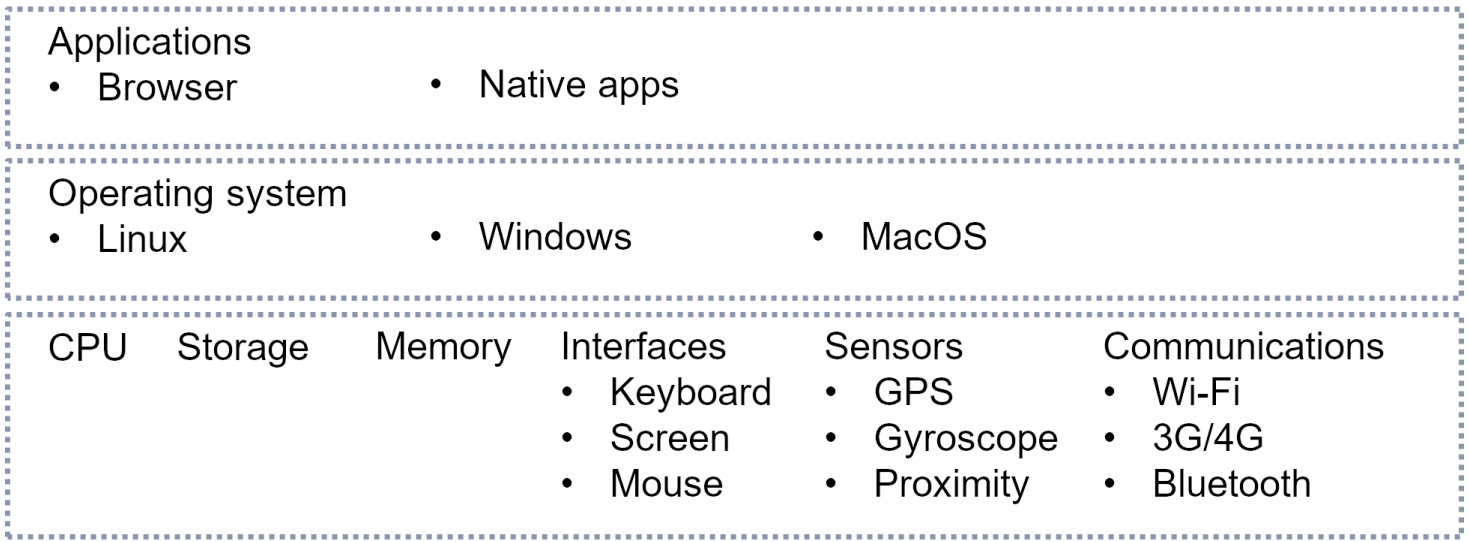
Web Mining is a subset of Internet Mining focusing on data generated by Internet applications. In many cases, mining tasks for Web applications might analyse data generated by lower layers too.

Technology Review

Internet devices

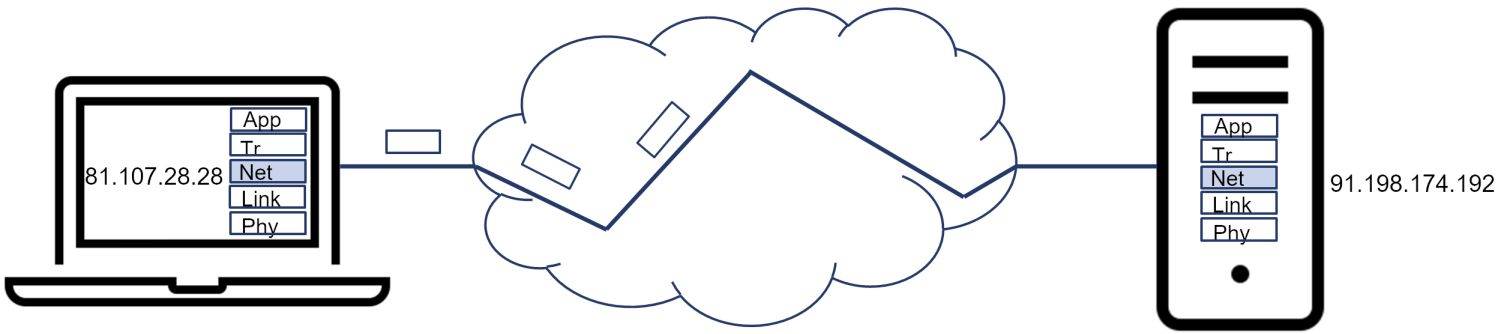
The data sources

Devices consist of hardware and software (typically operating system + application programs). Some devices are part of the Internet infrastructure (routers, switches), while others are connected as end-devices. All of them can generate Internet data.



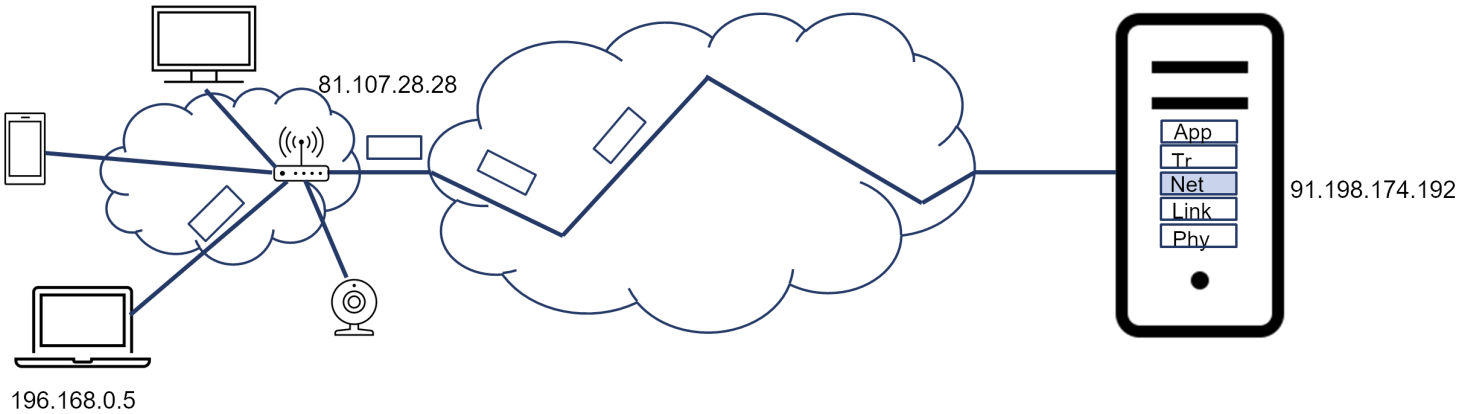
The Internet Protocol

Devices connected to the Internet have an IP address. The network layer breaks data into packets known as datagrams, that include the source and destination IP addresses and are routed over the internet.



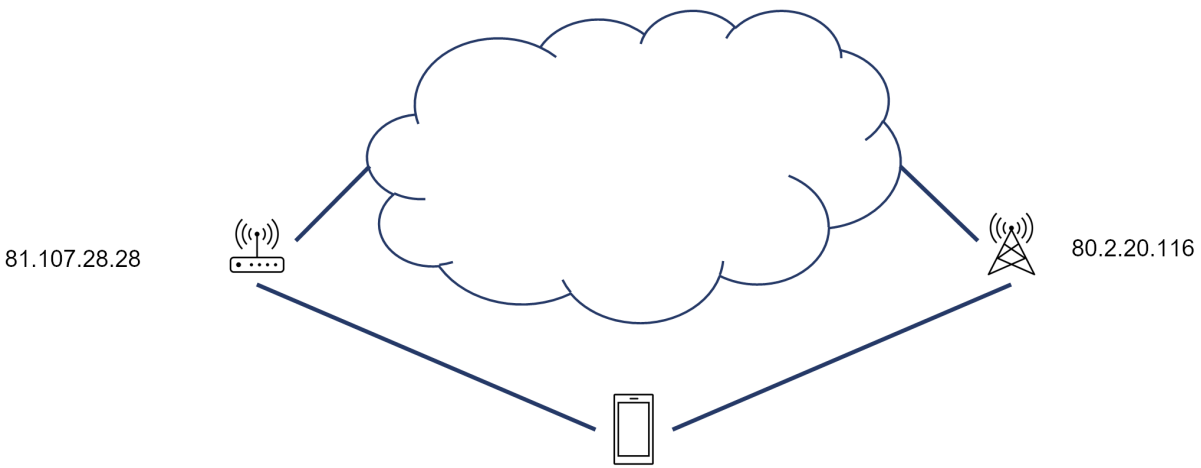
Private subnets: One public IP address, several devices

Devices connected to private subnets (e.g. home) have a private IP address and share the same public IP address (i.e. the router's). Network address translation (NAT) identifies the destination device for incoming datagrams, which use the public IP address.



Private subnets: One device, several public IP addresses

The same device might be associated to different public IP addresses at different times, as it can use different access points (e.g. one time a home Wi-Fi router, and another the cellular network).



HTTP messages

The Web consists of objects addressable by a URL, a string formed as "hostname/pathname", for instance

`www.someschool.edu/someDepartment/picture.gif`

HTTP is a communications protocol used to request and send such objects. HTTP messages are ASCII human-readable text and consist of a start line (type of message, URL, version, status) + headers (meta-data) + body (optional data).

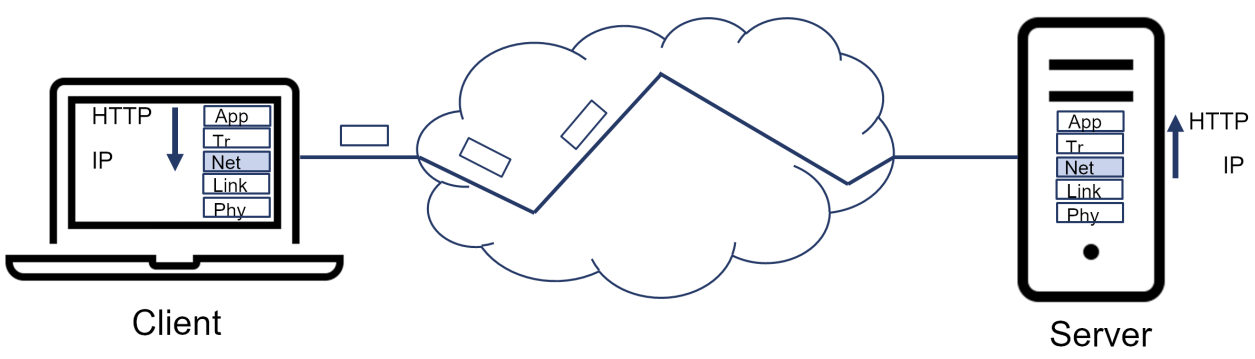
```
GET /someDepartment/picture.gif HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows
NT)
Host: www.someschool.edu
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

(Example GET request)

Clients and servers

HTTP uses a client-server architecture: The server receives HTTP requests from clients and sends HTTP responses back. Clients do not communicate directly with each other. HTTP messages are broken into IP datagrams at the source and are reassembled back at the destination.

Web browsers are applications that implement an HTTP client and display data received in HTML format.



API

An Application Programming Interface (API) defines how two software components interact. In the context of the Web, servers can offer a remote service by implementing it internally and exposing an API that can be accessed over the Internet by using HTTP.

API-based services include remote processing services (such as displaying a map, text analysis, face detection), and data collection.

For example, the Twitter API enables us to retrieve Twitter social data such as tweets, users, direct messages, lists, trends, media, and places.

Cookies

HTTP is stateless: servers are unable to keep track of their interactions with clients. However, sometimes it is desirable to keep track of the activity of a user (e.g. logins or shopping carts).

Cookies are small pieces of data provided by the server and stored at the client. The client sends them back with every requests, allowing the server to identify the client. Cookies are mostly used for:

- Session management (logins, shopping carts, game scoring)
- Personalisation (user preferences, themes)
- Tracking (recording and analysing user behaviour)

First-party cookies are set by the server to which requests are sent, while third-party cookies belong to other servers, which use them for advertising and tracking across the web.

Internet Mining Applications

Web data mining

Making sense of the web

Web search engines use information retrieval techniques to find contents in the web, such as web pages, images, videos and other files.

Search engines use web crawlers to visit web pages, that are subsequently parsed and indexed. When querying, results are retrieved based on the comparison between the query terms and parsing results, together with a quality metric based, among other factors on link analysis.

Unlike web crawling which parses and indexes pages, web scraping is the process of extracting data from websites or pages.

Social media analysis

Internet users as social beings

Online social media are platforms that allow Internet users to interact with each other and include online **social networks** (Twitter, LinkedIn, Facebook), **messaging** (WhatsApp, WeChat), **email** and **media sharing** platforms (YouTube, Flickr, Pinterest).

Users and their interactions can be mathematically modelled as a **graph**. By retrieving interactions between users (for instance, by using APIs exposed by social network platforms), graph and statistical methods can be used to:

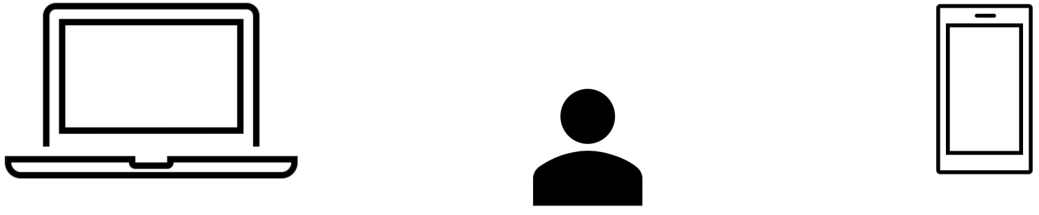
- Discover user communities
- Create recommendation systems
- Influence analysis

Identifying users

Multiple users, multiple devices, multiple touchpoints

From an economics point of view, the Internet is a **two-sided market**, where “free” services are paid by advertisers. Advertisers, in turn, are interested in **tracking** Internet users’ journeys, however these journeys usually involve multiple devices and channels (touchpoints).

Identity resolution involves associating Internet activity with individuals. Strategies for user identification include **third-party cookies**, **device identification**, **browser fingerprinting** and **traffic analysis**.



Attribution modelling

Who gets the credit?

A conversion in marketing is defined as a potential customer becoming an actual customer. Marketers need to measure the impact on specific marketing strategies on customer conversions.

Attribution modelling uses customer journeys generated by tracking users across multiple platforms, devices and channels, to measure the impact of each marketing strategy and determine their value.

- Rule-based attribution assigns conversion credit following a pre-specified model (first-touch, last-touch, uniform/linear, time-decay, U-shape...).
- Data-driven attribution includes other data (time from conversion, device type, number of ad interactions, order of ad exposure and creative assets).

Ingesting Internet data

Internet data sources

Where is my data coming from?

The internet can be seen as a huge and varied data source:

- Data is **sent** between devices across the internet
- Connected devices **store** data
- **Measure** the physical world
- **Interact** with users and **record** their activity

Data stored in a connected device or sent as the payload of a message are accompanied by metadata, which is data in its own right, for instance:

- EXIF image files contain geolocation information when taken by GPS-enabled cameras
- IP datagrams include the source and destination IP

Web data

There are two primary types of web data used by mining algorithms:

1. **Web content information**
   - Document data
   - Linkage data
2. **Web usage data**
   - Web transactions, ratings, and user feedback
   - Web logs

Harvesting methods

Harvesting data packets

In addition to data, packets sent over a network contain metadata that can be harvested. Network analysers or sniffers intercept and log traffic flowing through a network. This includes capturing wireless traffic (such as Wi-Fi or Bluetooth).

Network sniffers can extract the values of the fields in a packet identify (metadata), for instance source and destination IP addresses in datagrams. **Wireshark** is one of the most popular open-source sniffers.

Harvesting methods

Harvesting web servers

Web scraping is an automatic method to harvest data from webpages. This is accomplished by a program known as a bot, who uses HTTP to request data (HTML and other files) from a web server and then parses them.

Bots can also make HTTP requests to web servers via an **API**, who return data typically in JSON, hence retrieved data is well-formatted. Although similar, using APIs is not considered web scraping.

When APIs exist, the retrieved data might differ from the data obtained by web scraping or be insufficient because of request limits. Both methods should be considered complimentary.

Harvesting methods

Harvesting devices

Connected devices store data generated by the device as well as data describing its hardware and software components.

Device and **web browser information** can be inferred from **HTTP headers**. For instance, the user-agent header contains information about application type, operating system, device type and SW versions.

Applications running on a device can also obtain device information. For instance, **JavaScript AJAX** can fetch information about the device's screen resolution, time-zone and browser plugins and versions. Devices also have unique identifiers, such as the IDFA/AAID device ID, IMEI number and the address of its network interfaces (e.g. MAC address).

Harvesting methods

Harvesting users

Information about Internet users can be stored in their devices and in servers whose services they use (email, social networks, etc). This information includes **facts**, **preferences** and **behaviour**.

Although in decline, **HTTP cookies** are one of the most widely-used methods to store information about the user's preferences and past actions. The same third-party cookie can be associated to many different websites, hence users' browsing history can be reconstructed by the third party.

Apps can also be developed using libraries that track **click**, **tap**, **scrolling**, **mouse movements** (e.g. Hotjar), **visual focus** (e.g. SearchGazer), **record keystrokes**, entire contents of **visited webpages**, **screen**, **sounds**, **images**, **videos** and other **sensing data**.

Who harvests Internet data?

The actors behind the Internet data ecosystem

Users and devices connected to the Internet (as end-devices or part of the network infrastructure) generate Internet data. Internet data can be used to **understand** the Internet ecosystem, **monitor** service performance and quality (connectivity or application services), **create** new services and improve existing ones and for surveillance.

Examples of Internet data harvesters include:

- The **scientific community** and other groups harvest data for research purposes.
- **First-party** service providers collect data directly from their platforms.
- **Third-party** entities collect data from other platforms. Data brokers used harvested data, create new services that are sold to other companies.

Crawler Algorithms

Web crawlers iteratively find and fetch web links starting from an initial set of web addresses. The design of a real-world crawler is complex, with a distributed architecture and many processes or threads.

A basic crawler algorithm uses the following as input:

- An initial set of Universal Resource Locators (URLs) S
- A selection algorithm $\mathcal{A}$

Algorithm $\mathcal{A}$ decides which document to crawl next from a current frontier list of URLs, extracted from web pages.

Crawler Algorithms

Algorithm *BasicCrawler*(Seed URLs: S, Selection Algorithm: $\mathcal{A}$)
begin
  $FrontierList = S$;
  repeat
    Use algorithm $\mathcal{A}$ to select URL $X \in FrontierSet$;
    $FrontierList = FrontierList - \{X\}$;
    Fetch URL X and add to repository;
    Add all relevant URLs in fetched document X to
      end of $FrontierList$;
  until termination criterion;
end

Crawler Algorithms

The choice of the selection algorithm $\mathcal{A}$ will typically result in a bias in the crawling algorithm.

Selection algorithms:

- Breadth-first: if new URLs are appended to the end of the frontier list and $\mathcal{A}$ selects documents from the beginning of the list.
- Depth-first: the crawler will travel as far down one branch.
- Frequency-Based: Frequently updated pages are selected by $\mathcal{A}$.
- PageRank-Based: Choose Web pages with high **PageRank** from frontier list.

Search Engine Indexing & Ranking

The Process of Search

After the documents have been crawled, they are leveraged for query processing. There are two primary stages to the search index construction:

1. **Offline stage**
   - The search engine preprocesses the crawled documents to extract the tokens and constructs an index.
   - A quality-based ranking score is also computed for each page.
2. **Online query processing**
   - The relevant documents are accessed and then ranked using both their relevance to the query and their quality.

Ranking

Ranking is typically based on a content-based score for each document:

- A word is given different weights, depending upon whether it occurs in the title, body, URL token, or the anchor text.
- The number of occurrences of a keyword in a document will be used in the score.
- The prominence of a term in font size and color may be leveraged for scoring.
- When multiple keywords are specified, their relative positions in the documents are used as well.

Limitations of Content-Based Score

- It does not account for the reputation, or the quality, of the page.
- Web Spam
  - Content-spamming: The Web host owner fills up repeated keywords in the hosted Web page
  - Cloaking: The Web site serves different content to crawlers than it does to users.
- Search Engine Optimization (SEO)
  - The Web set owners attempt to optimize search results by using their knowledge.

Ranking

- Reputation-Based Score
  - Page citation mechanisms: When a page is of high quality, many other Web pages point to it.
  - User feedback or behavioral analysis mechanisms: When a user chooses a Web page, this is clear evidence of the relevance of that page to the user.
- The Final Ranking Score

$$RankScore = f(IRScore, RepScore)$$

where *IRScore* and *RepScore* are the content- and reputation-based scores respectively, and $f(\cdot)$ is a (typically proprietary) function used by the search engine.

PageRank

The PageRank algorithm uses the linkage structure of the Web for reputation-based ranking.

It precomputes the reputation portion of the score, and is therefore independent of the user query.

The basic idea is that highly reputable documents are more likely to be cited (or in-linked) by other reputable Web pages.

Random Walk Model

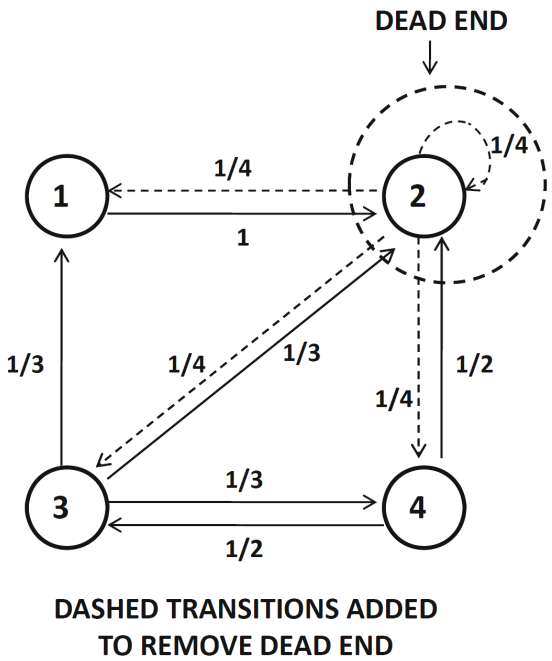
A random surfer who visits random pages on the Web by selecting **random** links on a page:

- The long-term relative frequency of visits to any particular page is clearly influenced by the number of **in-linking** pages to it.
- The long-term frequency of visits to any page will be higher if it is linked to by other **frequently** visited (or reputable) pages.

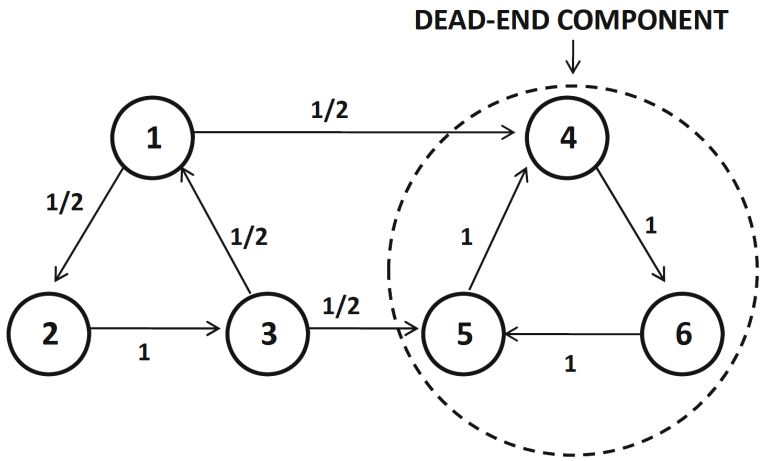
This model is also referred to as the random walk model. This long-term frequency is also referred to as the steady-state probability.

Issues with Random Walk Model

- Dead ends: pages with no outgoing links
- Dead-end components: groups of nodes with no outgoing links.



(a) Dead-end node



(b) Dead-end component

Solutions for Random Walk Model

- Dead End: add links from the dead-end node (Web page) to all nodes (Web pages), including a self-loop to itself.
- Dead-end components: A teleportation (restart) step: The random surfer may either jump to an arbitrary page with probability α, or it may follow one of the links on the page with probability $(1 - \alpha)$.

PageRank

The PageRank of node i is equal to the **probability of transition into node** i. This is defined by the sum of the probabilities of the teleportation and transition events:

$$\pi(i) = \alpha/n + (1-\alpha) \cdot \sum_{j \in In(i)} \pi(j) \cdot p_{ji}$$

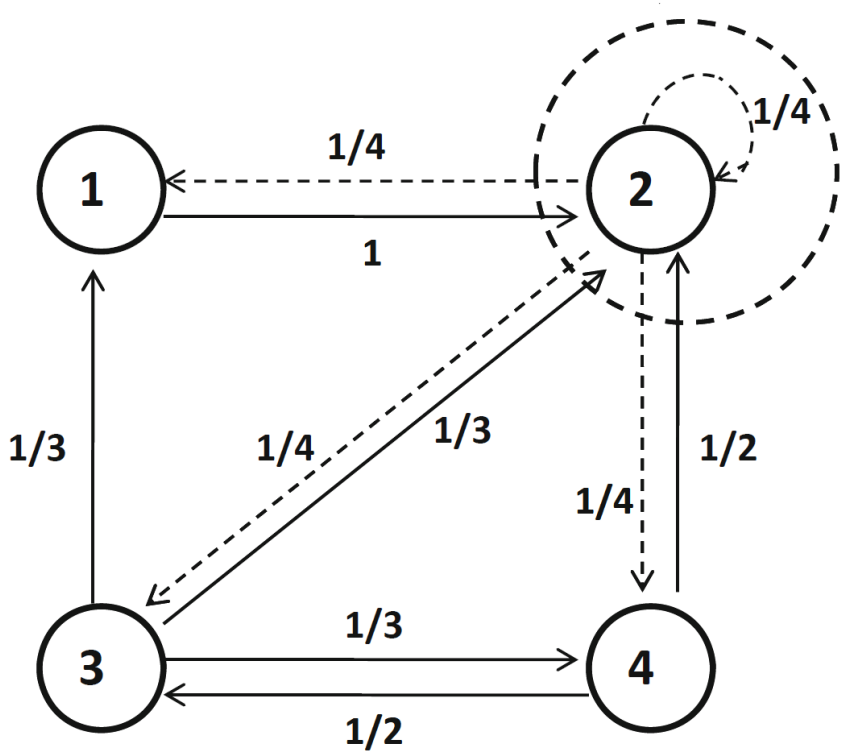
where:

- n is the total number of nodes
- $In(i)$ is the set of nodes incident on i
- α/n is the probability of a teleportation into i
- $p_{ji} = 1/|Out(j)|$ is the probability of transitioning from node j to node i
- $(1-\alpha) \cdot \sum_{j \in In(i)} \pi(j) \cdot p_{ji}$ is the probability of a transition into i
- $Out(i)$ is the set of end points of the outgoing links of node i.

PageRank

The PageRank for node 2 in the below example can be computed as follows:

$$\pi(2) = \alpha/4 + (1-\alpha) \cdot \big(\pi(1) + \pi(2)/4 + \pi(3)/3 + \pi(4)/2\big)$$



PageRank

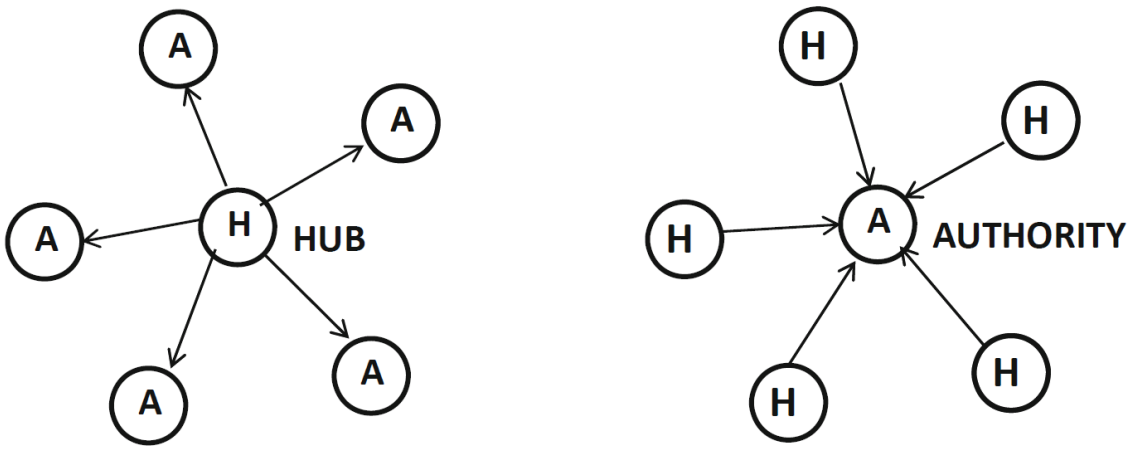
- Having all equations for all nodes, the PageRanks can be estimated iteratively as a linear system of equations, using the power-iteration method.
- PageRank computation is expensive, and it cannot be computed on the fly for a user query during Web search.
- Many PageRank extensions and optimisations exist, such as Topic-Sensitive PageRank and SimRank.

PageRank implementation for very large graphs: https://github.com/louridas/pagerank

HITS

The Hypertext Induced Topic Search (HITS) algorithm is a query-dependent algorithm for ranking pages.

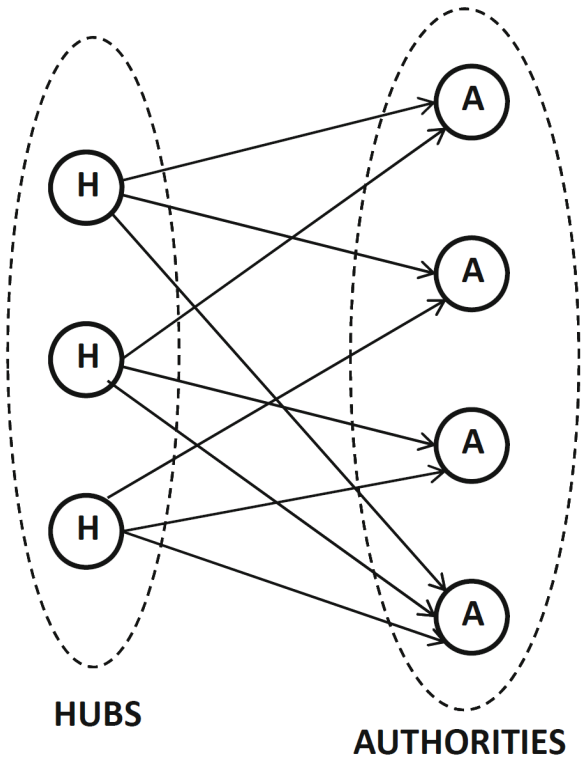
The intuition behind HITS lies in an understanding of the typical structure of the Web that is organized into hubs and authorities.



- Authority: A page with many in-links.
- Hub: A page with many out-links to authorities.

The insight of HITS

- Good hubs point to many good authorities.
- Good authority pages are pointed to by many hubs.



HITS Algorithm

HITS starts with the list of relevant pages and expands them with a hub ranking and an authority ranking:

1. Collect the top-r most relevant results to the search query. This defines the root set R.
2. For each node in R, the algorithm determines all nodes (in-linking or out-linking) immediately connected to R, called the base set S.
3. Let $G = (S, A)$ be the subgraph defined on S, where A is the set of edges between nodes in the root set.

HITS implementation: http://search.cpan.org/ xern/Algorithm-HITS-0.04/

HITS Algorithm

4. Each page (node) $i \in S$ is assigned both a hub score $h(i)$ and authority score $a(i)$:

$$h(i) = \sum_{j:(i,j)\in A} a(j) \quad \forall i \in S$$

$$a(i) = \sum_{j:(j,i)\in A} h(j) \quad \forall i \in S$$

The basic idea is to reward hubs for pointing to good authorities and reward authorities for being pointed to by good hubs.

The above forms a linear system of equations that can be solved iteratively, producing estimated hub and authority vectors for all nodes in S.

Questions?
also please use the forum on QM+