



UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

BIG DATA AND DIGITAL METHODS

LECTURE 2

APPLICATIONS AND TOOLS

Dario Malchiodi

SOME APPLICATIONS

- Finding similar items
- Link analysis
- Market-basket analysis
- Web advertising
- Recommendation systems

FINDING SIMILAR ITEMS

- in search of plagiarism
- to improve the efficiency of search engines and news aggregators
- when dealing with messy data sets potentially containing different versions of the same information (entity resolution)
- in order to recommend products

FINDING SIMILAR ITEMS

Requires

- Define a measure of similarity between items
- Finding a suitable representation for each item (e.g., when working with textual information, removing stop words, stemming, shingling, ...)
- Efficiently store that representation
- Summarize the available information (preserving similarity)

SIMILARITY MEASURES

- Euclidean distance(s)
- Cosine distance
- Jaccard distance
- Edit distance

LINK ANALYSIS

- Finding the most important Web pages within the result of a search query
- Crucial in order to build an efficient search engine
- Typical example: PageRank

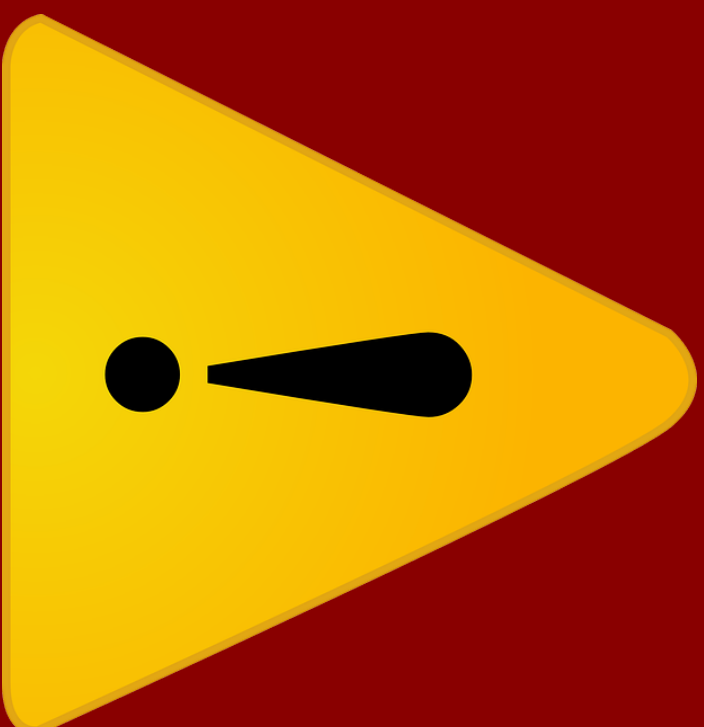
PAGERANK

- One of the ingredients of Google's search engine
- Informally: a Web page is important when is linked by other important pages

PAGERANK

More formally: simulate a population of *random surfers*

- Initially the population is uniformly distributed on all Web pages
- Each random surfer has the same behaviour: it selects uniformly at random one of the links in the page and follows it, then repeats the process on the new page, and so on
- It can be shown that the probability distribution of the surfers on the Web pages will converge to a fixed result
- The higher this fixed probability for a page, the more important that page will be considered



Warning: entering math zone!

MATRICES AND LINEAR APPLICATIONS

- A *linear application* between $\mathbb{R}$ and $\mathbb{R}$ has the general form
$$f(x) = \alpha \cdot x$$
- A linear application between vectors in $\mathbb{R}^m$ has the general form

$$f(v) = Mv,$$

where $v \in \mathbb{R}^m$ is a vector of m elements and M is a $m \times m$ matrix

- and

$$(Mv)_i = \sum_{j=1}^m m_{ij}v_j$$



You are safely leaving math zone!

PAGERANK

Even more formally, given

- the *transition matrix* M of the Web (m_{ij} is 0 if page i does not contain a link to page j and $1/n$ otherwise, where n is the number of outgoing links from i)
- a vector v_0 whose components are initially all set to $1/n$, being n the number of Web pages

The limiting distribution of random surfers is obtained through iteration of the following equation

$$v_{t+1} = Mv_t$$

PAGERANK

This is a big data problem:

- M is a huge square matrix (the order of magnitude of m is at least ten billions)
 - although it is *sparse*, i.e. most of its entries are null
- Solution: store M through Hadoop and write a MapReduce procedure computing Mv

COMPUTING Mv

- store M as (i, j, m_{ij}) tuples
- (taking advantage of the sparseness)
- map:
 - stores v locally
 - input: (i, j, m_{ij})
 - output: key i , value $m_{ij}v_j$
- reduce:
 - input: key i , value $[m_{i1}v_1, \dots, m_{im}v_m]$
 - output: key i , value $\sum m_{ij}v_j = (Mv)_i$

MARKET-BASKET ANALYSIS

(OR FREQUENT ITEMSETS SEARCH)

- given a set of *baskets* (typically, the receipts emitted by a brick-and-mortar or online retailers)
- find out if two (or more) items tend to be sold jointly
- to build marketing campaigns (e.g. a sale on hot dogs after the mustard price has been raised)
- to send vouchers to clients buying only some of the items in the frequent itemset)

MARKET-BASKET ANALYSIS

Not just for marketing...

- Finding related concepts in documents: items → words, baskets → documents
- Detecting plagiarism: items → documents, baskets → sentences
- Genomic analyses: items → biomarkers and diseases, baskets → all data about a patient (genome analyses and medical history of disease)

MARKET-BASKET ANALYSIS

It's a big data problem

- even focusing on frequent pairs, given n items there are

$$\binom{n}{2} \approx \frac{n^2}{2}$$

possible pairs to be checked

- a major chain sells $n = 100000$ items and should consider millions of market baskets

MARKET-BASKET ANALYSIS

Solution: the obvious approach (scan baskets and count pairs), although with a two-pass scan on baskets in order to avoid considering all possible pairs, and efficiently exploiting the PC central memory

Or more sophisticated algorithms, even approximate ones: at the end of the day, you don't want all frequent itemsets, only as much as you need

WEB ADVERTISING

Traditional media: print the ad on a newspaper, with a cost per impression, reaching a high number of people (mostly uninterested)

Some cleverness: use specialised press for specific market niches (e.g. place ads for golf clubs in a golf journal)

WEB ADVERTISING

Digital media: use information about each specific reader in order to display (hopefully) relevant ads

- the user may belong to a golf-related group on a social network
- the frequency of the word “golf” in the user mail messages is sensibly higher than the average
- the user may have queried a search engine for golf-related terms
- the user may browse frequently golf pages on the Web, or even have them stored in his/her bookmarks
- privacy warning here!

WEB ADVERTISING

How to decide which ads place, for instance, in a Google page showing the results of a search query?

- only a limited number of ads can be placed in a page
- the advertiser pays “per click” (rather than “per impression”)
- the advertiser has a budget fixed over a time period
- an ad should be associated to the results of search queries containing one or more meaningful terms (e.g., golf)

If one knows in advance, say, the next 1000 queries to the search engine, it is easy to assign ads in order to maximize the revenue. That is not (obviously) the case, thus it is necessary to resort to approximate algorithms

RECOMMENDER SYSTEMS

A brick-and-mortar retailer has a (relatively small) number of items

Recommendation of products is typically related to the most sold items (say, the ten most popular books in a library, possibly “projected” on genres)

RECOMMENDER SYSTEMS

An online store sells a very high number of items

If only the most sold items get recommended, most of the possible revenue is lost (the so-called “long-tail” phenomenon)

Thus recommendations should be more personalized for each

user

RECOMMENDER SYSTEMS

- Typical approach: collaborative filtering
- Find groups of similar users: two users are similar if they happen to buy the same items
- Suggest a user u items that users similar to him/her have bought and he/she hasn't

SOME TECHNIQUES

- Clustering
- Machine learning
- Dimensionality reduction

CLUSTERING

- Input: a set of objects, each described by a vector of values
- Goal: finding a grouping of objects, that is
 - a set of *clusters*
 - with each object belonging to exactly one cluster
 - objects in a same cluster are “similar”
 - objects in different clusters are “dissimilar”
- Key problems:
 - finding out a distance between objects
 - discovering how many clusters

CLUSTERING

Variations:

- overlapping clusters
- fuzzy clusters: each object belongs to each cluster with different *membership values*

MACHINE LEARNING

- Usually: problem, solution, algorithm, program
- Pitfalls: no solution is known, complexity issues
- In such cases: approximate solutions
- One possibility: learning from data

MACHINE LEARNING

- Sometimes it's easy to solve a problem...
- ...although it's difficult to explain how the solution was found (e.g., face recognition)
- sample of problem instances and solutions → approximate solver
- several approaches:
 - neural networks
 - support vector machines
 - naive bayes
 - logistic regression
 - C4.5/C5.0
 - and many more...

DIMENSIONALITY REDUCTION

- Each data item is typically encoded in a vector
- e.g., a picture is a sequence of pixel, in turn encoded as a RGB numeric triple
- What if this vector has a big dimension...
- ...but it contains redundancy or uninformative components?
- e.g., an object centered in the picture or showing objects with given regularities
- $\Rightarrow$ compression of each vector in order to summarize most of the contained information

DIMENSIONALITY REDUCTION

- several approaches:
 - principal component analysis (PCA)
 - independent component analysis (ICA)
 - singular value decomposition (SVD)
 - and many more...

