# Advanced Topics in Network Analysis
## Link prediction, Recommender Systems

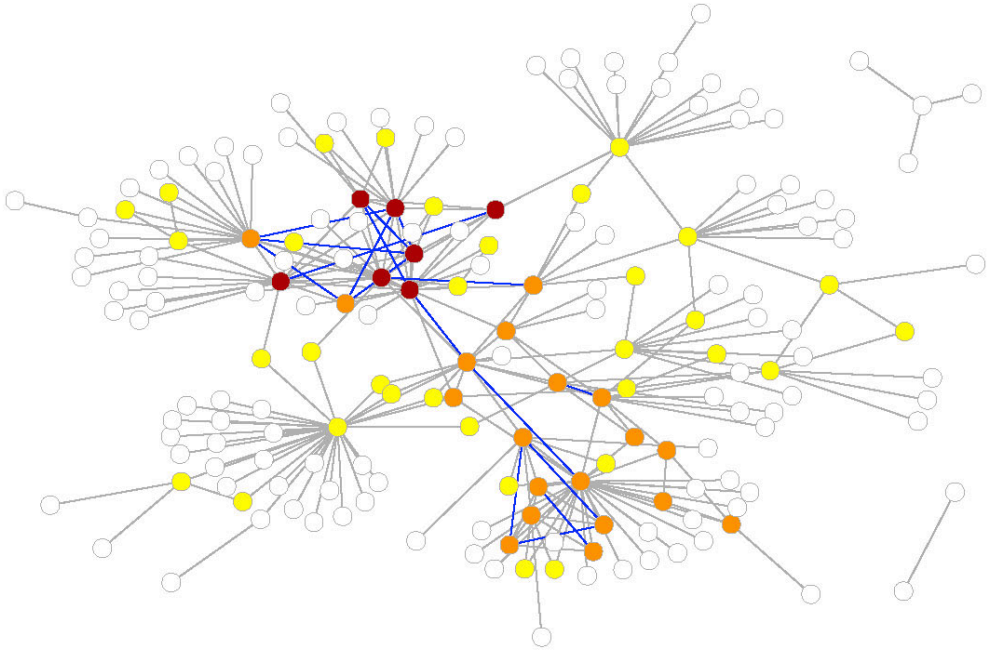
Carlo PICCARDI

DEIB - Department of Electronics, Information and Bioengineering
Politecnico di Milano, Italy

email carlo.piccardi@polimi.it
http://home.deib.polimi.it/piccardi

# LINK PREDICTION

Networks are used to model interactions observed in field/laboratory experiments, or obtained through data processing, etc.



Many existing interactions might be overlooked (also, a few observed interactions might be artificial).

Main assumption: the observed network $A^O$ is a "noisy" observation of a true network $A^{true}$.

Problem: reconstruct $A^{true}$ given $A^O$.

Applications:

- find missing links (perhaps suggesting ad-hoc experiments)
- delete spurious links
- in time-varying networks, predict future links (e.g., who will be your future friends?)

## Finding missing links: Similarity-based algorithms

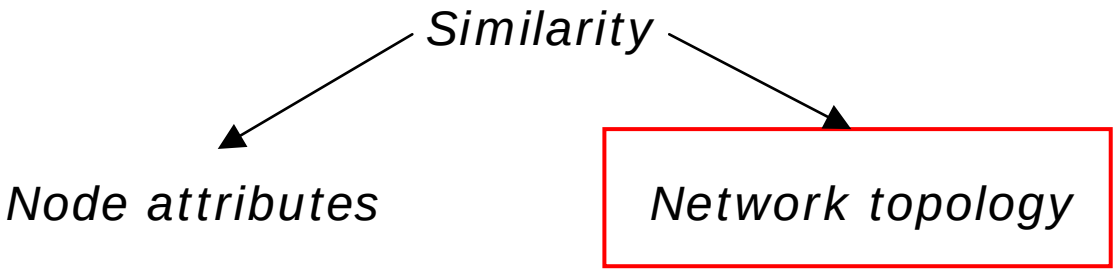
The observed network $A^O$ (undirected, unweighed) has $N$ nodes ($i = 1,2,\dots,N$) and $L$ observed links ($a_{ij} = 1$).

$S$ is the set of non observed links ($a_{ij} = 0$), whose cardinality is $\left(\frac{N(N-1)}{2} - L\right)$.

Assumption: the set $S$ contains missing links (=existing but not observed).

*The more two nodes are "similar", the more is the likelihood they are connected*

- define a similarity score $s_{ij}$ for each node pair $(i,j)$
- define a threshold $\bar{s}$
- a non observed link is a missing link if $s_{ij} > \bar{s}$
- *(optional: an observed link is a spurious link if $s_{ij} < \bar{\bar{s}} < \bar{s}$)*

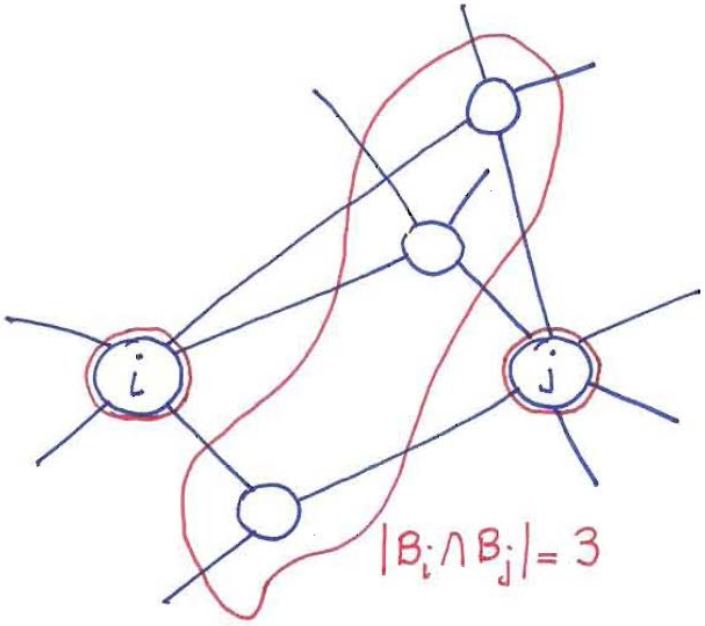


How to quantify the structural (topological) similarity of nodes $(i,j)$ ?

- Number of common neighbours: $s_{ij} = |B_i \cap B_j| = (A^2)_{ij}$ = *n. of length-2 paths*
- … many variations (normalizations e.g. by functions of the degrees $k_i$,$k_j$)
- generalization: consider paths $(i,j)$ longer than 2 (Katz index)

$$s_{ij} = \beta A_{ij} + \beta^2 (A^2)_{ij} + \beta^3 (A^3)_{ij} + \cdots = [(I - \beta A)^{-1} - I] \quad , \quad 0 < \beta < 1/\lambda_{max}(A)$$

- … Katz index truncated at some power
- Preferential attachment index: $s_{ij} = k_i k_j$
- Resource allocation index: $s_{ij} = \sum_{h \in (B_i \cap B_j)} {}^{1}/_{k_h}$
- …many others

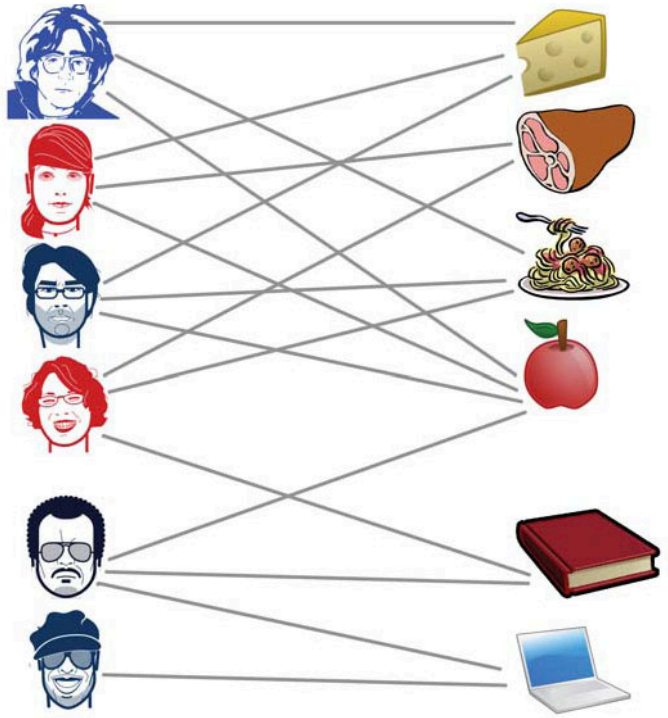


# RECOMMENDER SYSTEMS

$U = \{u_1, u_2, \ldots, u_m\}$ : set of users $\qquad$ $O = \{o_1, o_2, \ldots, o_n\}$ : set of objects

The set of objects owned by each user $u_i$ is coded by the bipartite network $B = [b_{ij}]$:

$b_{ij} = 1$ if $u_i$ owns $o_j$, $\qquad$ $b_{ij} = 0$ otherwise



- user degree (number of objects owned by $i$):

$$k(u_i) = \sum_j b_{ij}$$

- object degree (number of users owning $j$):

$$k(o_j) = \sum_i b_{ij}$$

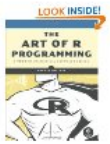
Problem: assuming $u_i$ likes the objects she/he owns (=not purchased at random), predict (and recommend) which more object(s) she/he could like.

## Customers Who Bought This Item Also Bought

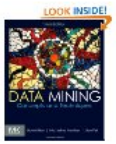
Page 1 of 15

Programming Collective Intelligence: Building ...
› Toby Segaran
(84)
Paperback
$26.39

Mining the Social Web: Analyzing Data from ...
› Matthew A. Russell
(19)
Paperback
$26.36



The Art of R Programming: A Tour of Statistical ...
Norman Matloff
(29)
Paperback
$24.99

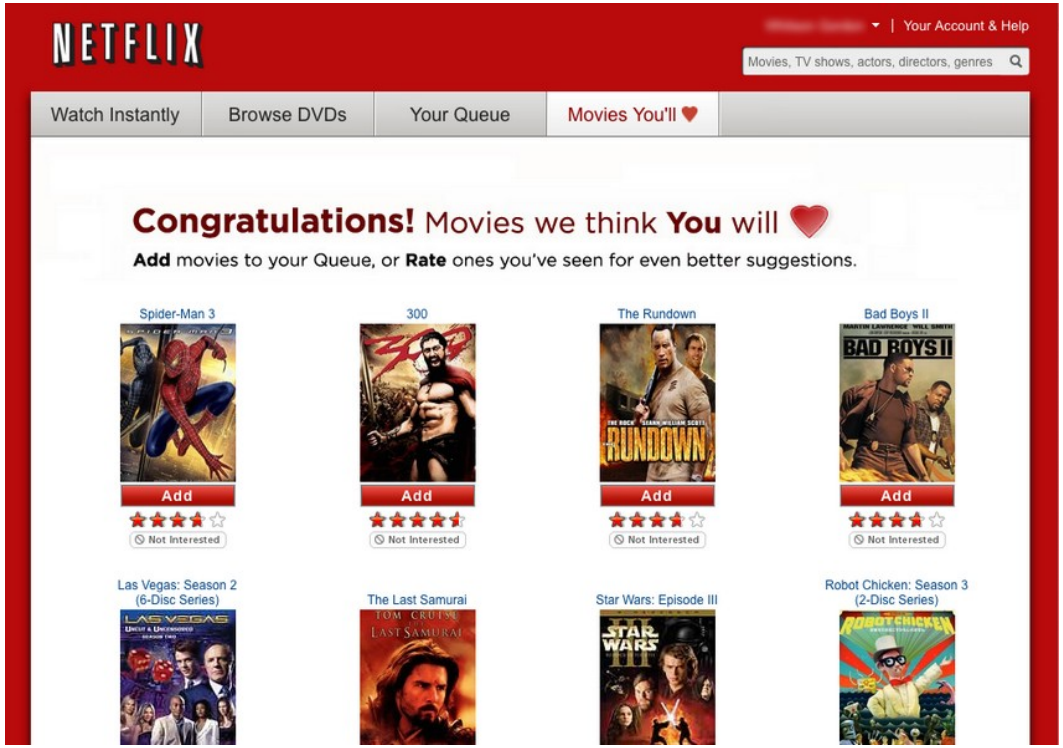


Data Mining: Concepts and Techtechniques, Third ...
› Jiawei Han
(12)
Hardcover
$51.99

The Elements of Statistical Learning: Data Mining, ...
› Trevor Hastie
(46)
Hardcover
$65.96

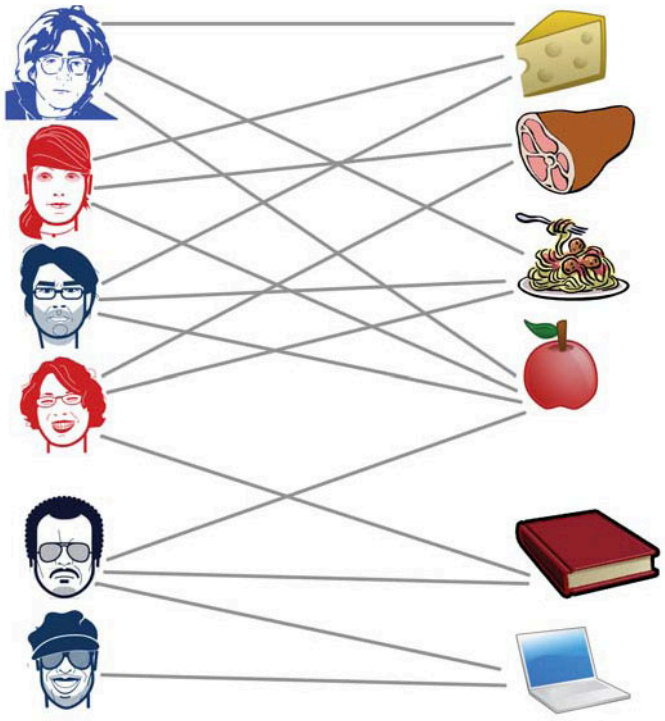
Introduction to Data Mining
Pang-Ning Tan
(21)
Hardcover
$101.39

Data Mining with R: Learning with Case ...
Luis Torgo
(5)
Hardcover
$67.70




A naive solution ("global ranking method"):

recommend the objects $o_j$ with the largest degree $k(o_j)$
(=the object owned by the largest number of users).



The recommendation list is the same for all users (no personalization):

*1st:* apple ($k = 4$)

*2nd:* cheese, spaghetti ($k = 3$)

*4th:* ham, book, laptop ($k = 2$)

The network is not used.

Personal recommendation systems provide a different ("personalized") list of objects to each user $u_i$.

Main assumption: similar users like similar objects ("collaborative filtering").

The score assigned to the (non-existing) pair $(i,j)$:

$$\hat{b}_{ij} = \frac{s_{i1}b_{1j} + s_{i2}b_{2j} + \cdots + s_{im}b_{mj}}{s_{i1} + s_{i2} + \cdots + s_{im}} = \frac{\sum_{l=1}^{m} s_{il}b_{lj}}{\sum_{l=1}^{m} s_{il}}$$

It is a link prediction procedure on a bipartite network.

How to quantify the similarity between users $, l$ ?

Two examples (among the simplest ones) of user similarity:

- topological similarity: common neighbours = n. of objects in common:

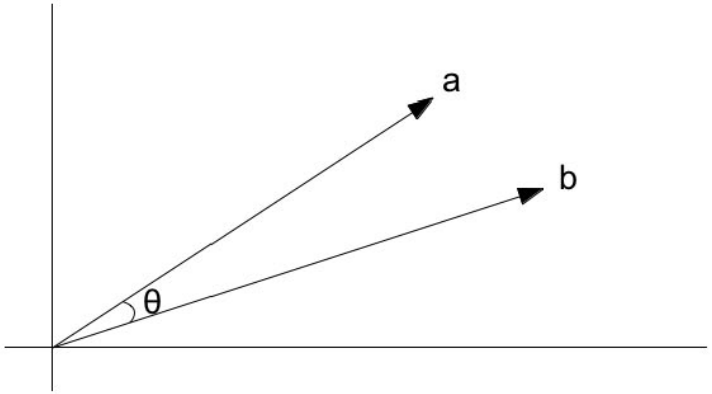
$$s_{il} = \sum_{j=1}^{n} b_{ij} b_{lj}$$

- rating-base similarity: users give a score (e.g., 1 to 5 stars) to the objects they own:

$$r_i = (r_{i1} \quad r_{i2} \quad \dots \quad r_{in})$$

$r_{ij}$ is the score given to object $j$ by user $i$.

The more the vectors $r_i, r_l$ are close each other (=same preferences), the more users $i, l$ are similar.




"Cosine" similarity:

$$s_{il} = \frac{r_i \cdot r_l}{|r_i||r_l|}$$