Tagging structure in a protein-protein interaction network, a co-authorship network and the (English) Wikipedia

Gergely Palla, Illés J. Farkas, Péter Pollner, Imre Derényi, Tamás Vicsek
Eötvös University and Hungarian Academy of Sciences (Budapest, Hungary)

Proteins
Math co-authors
Wikipedia articles

Network
Nodes + Links

Biochemical functions Classification of co-authored papers

Node tags
Directed Acyclic Graph
Growt
Cell growth
Combinatorics
Graph theory
FIFA World Cup
FIFA World Cup Players

Category tree

CFinder.org
Full version:

http://CFinder.org --> Publications

G. Palla, I. J. Farkas, P. Pollner, I. Derényi, T. Vicsek
Fundamental statistical features and self-similar properties of tagged networks
There is no clearly separated group of “most popular tags”
Transition between popular and less popular tags is continuous
Probability that a given tag and its descendants label a portion $x$ of all nodes
There is no clear group of “most heavily tagged nodes”

Transition between strongly and less strongly tagged nodes is continuous

Probability that a node has \( n \) tags

Number of tags \( (n) \) on a node (protein, author, wiki article) of the network
On the large scale there is no “link saturation” within a topic. In fact, link density within a large topic is almost the same as outside.

Number of links among these nodes

Number of Wikipedia articles selected from those labeled with “Japan” and descendant terms

Maximum possible

Wikipedia average

Number of Wikipedia articles selected from those labeled with “Japan” and descendant terms
Meaningful removal of loops from the category hierarchy
How to achieve tree structure (DAG) with the lowest number of removals

Example (Oct.2007):

Finding all loops is easy.

But which of their links should be removed?

**Goals:**
- Remove lowest possible number of links
- Smallest “damage” to existing category hierarchy
Meaningful removal of loops from the category hierarchy
How to achieve tree structure (DAG) with the lowest number of removals

Example (Oct. 2007):

(1) Identify “loop subgraph” by iteratively removing nodes with 1 link
(2) Iteratively remove “least important” links from the loop subgraph
(3) Add non-loop links again

Finding all loops is easy.

But which of their links should be removed?

Goals:
- Remove lowest possible number of links
- Smallest “damage” to existing category hierarchy
http://CFinder.org -- with network data and free analysis software

We thank for support from:

[Logos of NKTH, OIKA, and Magyary Zoltán Felsőoktatási Kőzalapítvány]