Random Walks

- Random Walks on graphs
  - Google's page rank

Google's PageRank

- Assumption: A **link** from page A to page B is a **recommendation** of page B by the author of A (we say B is *successor* of A)

➔Quality of a page is related to its in-degree

- Recursion: Quality of a page is related to
  - its in-degree, and to
  - the *quality* of pages linking to it

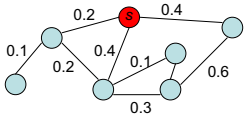
➔**PageRank** [BP '98]

Definition of PageRank

- Consider the following infinite random walk (surf):
  - Initially the surfer is at a random page
  - At each step, the surfer proceeds
    - to a randomly chosen web page with probability d
    - to a randomly chosen successor of the current page with probability 1-d
- **The PageRank of a page p is *the fraction of steps the surfer spends at p in the limit.***

Random walks **with restarts** on interaction networks

- Consider a random walker that starts on a source node, s. At every time tick, the walker chooses randomly among the available edges (based on edge weights), or goes back to node s with probability c.



Random walks on graphs

- The probability $p_s(v)^{(t)}$, is defined as the probability of finding the random walker at node v at time t.
- The steady state probability $p_s(v)$ gives a measure of affinity to node s, and can be computed efficiently using iterative matrix operations.

Computing the steady state **p** vector

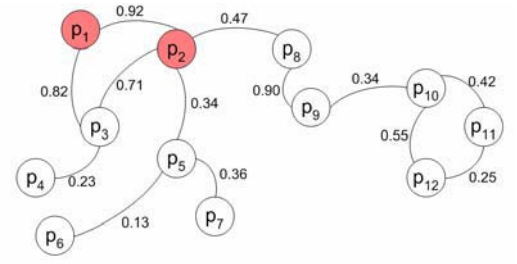
- Let **s** be the vector that represents the source nodes (i.e., $\mathbf{s}_i = 1/n$ if node i is one the n source nodes, and 0 otherwise).
- Compute the following until **p** converges:

$$\mathbf{p} = (1\text{-}c)\mathbf{A}\mathbf{p} + c\mathbf{s}$$

where **A is the column normalized adjacency matrix** and c is the restart probability.

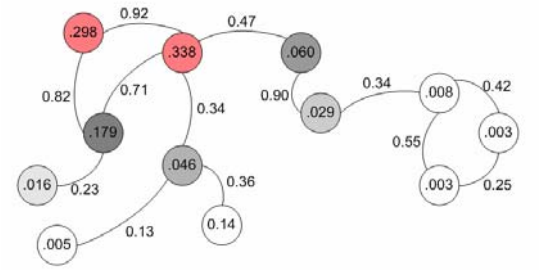
Same example

- Start nodes: p_1 and p_2



Random walk results

- Restart probability, c = 0.3



Experiments

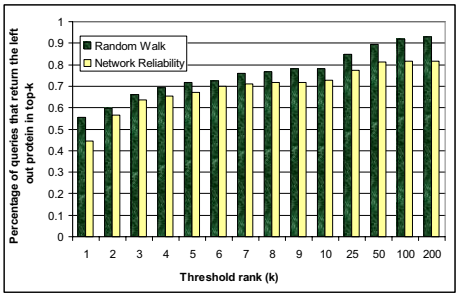
- Conducted complex/pathway membership queries on a probabilistic Yeast network:
  - ConfidentNet (Lee *et al.*, 4,681 nodes, 34,000 edges)
- Assembled a test set of 27 MIPS complexes and 10 KEGG pathways.

Leave-one-out benchmark

- Leave one member (in turn) from each complex/pathway.
- Use the rest of the complex/pathway as the starting, i.e., query, set.
- Examine the rank of the left-out protein.

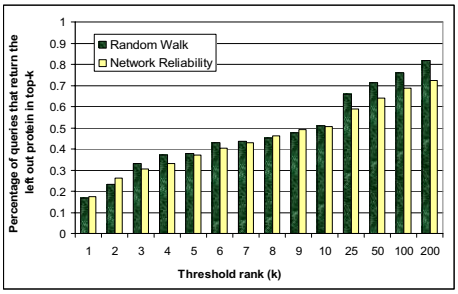
Leave-one-out on ConfidentNet

- MIPS complex queries



Leave-one-out on ConfidentNet

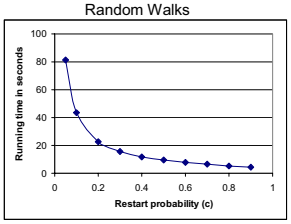
- KEGG pathway queries



Running time

- Total time to complete 121 MIPS complex queries

Random Walks



Network Reliability by Monte Carlo Sampling