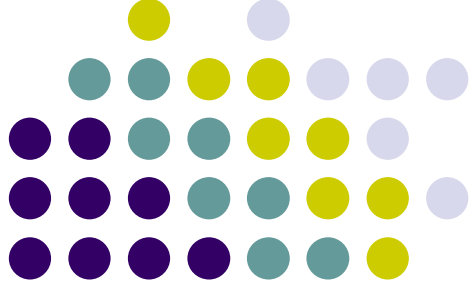


Graph similarity

Laura Zager and George Verghese
EECS, MIT

January, 2005

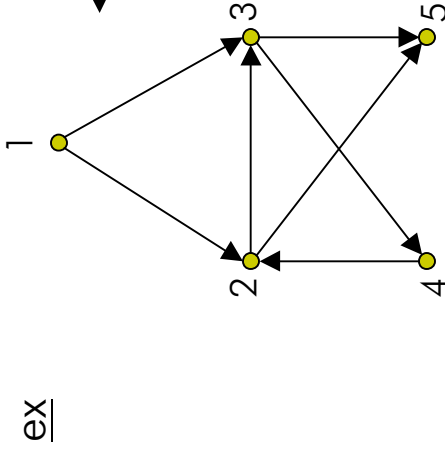


Some quick definitions

$G(V, E)$

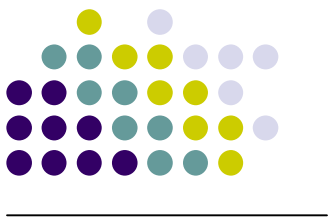
V \longleftarrow the set of vertices

$E \subset V \times V$ \longleftarrow the set of edges – can be directed or undirected.

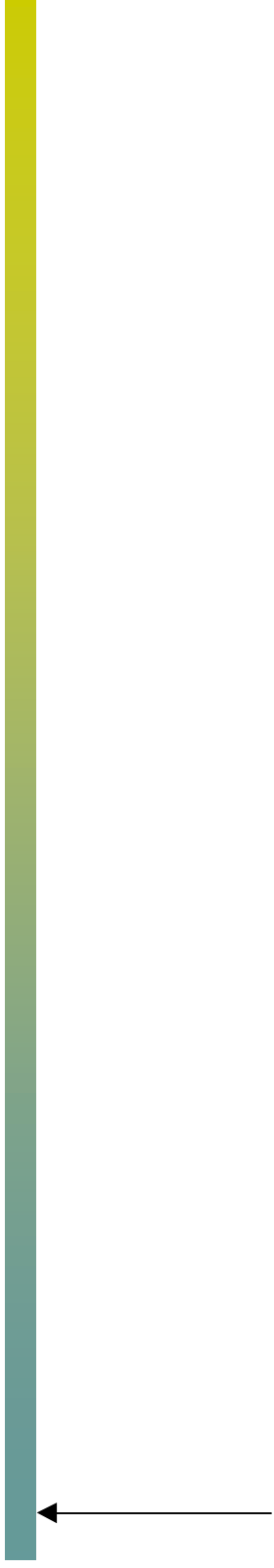
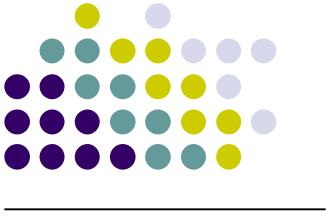


a directed graph and its
node-node adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

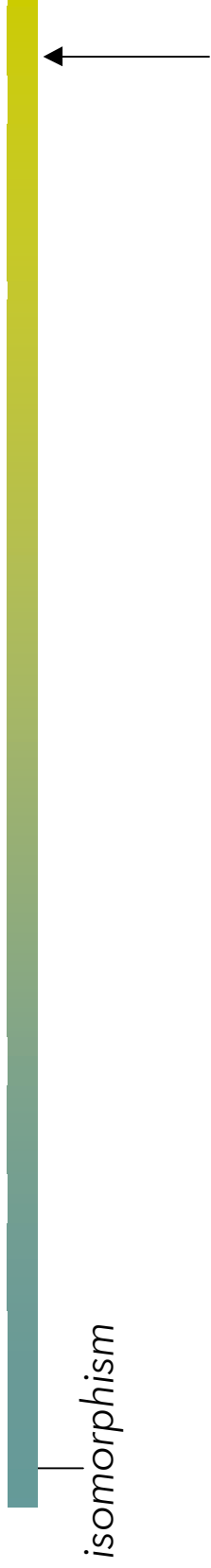
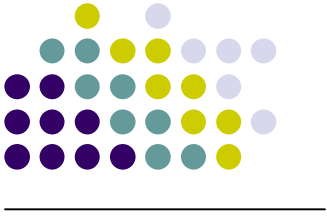


Notions of similarity



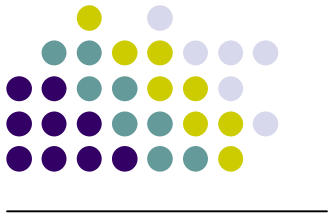
- Isomorphism – identifying a *bijection* between the nodes of two graphs which preserves (directed) adjacency.
- Corneil & Gottlieb, *Journal of the ACM*, 1970.
- Pelillo, *Neural Computation*, 1999.
- Ullman, *Journal of the Assoc. of Computing Machinery*, 1976.

Notions of similarity



- Statistical methods – assessing aggregate measures of graph structure (e.g. degree distribution, diameter, connectedness).
 - Albert, Barabasi, *Reviews of Modern Physics*, 2002
 - Dill, Kumar, et al., *ACM Transactions on Internet Technology*, 2002.
 - Watts, *Small Worlds*, 1999.

Notions of similarity



?



isomorphism

?

statistical
comparison

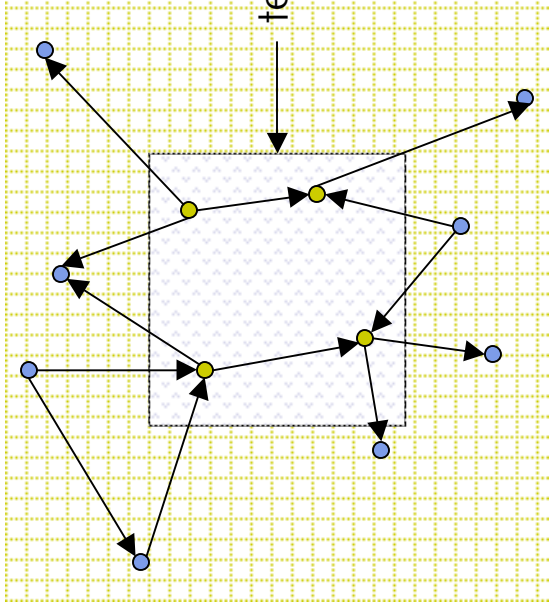
- Iterative methods:

Two graph elements (e.g., edges or nodes) from two different graphs are *similar* if their neighborhoods are *similar*.

- Kleinberg, *Journal of the ACM*, 1999. ←
- Blondel, Van Dooren, et al., *SIAM Review*, 2004. ←
- Jeh & Widom, *8th Intl. Conf. on Knowledge Discovery and Data Mining*, 2002. ←
- Melnik, Garcia-Molina, *18th Intl. Conf. on Data Engineering*, 2002.
- Heymans & Singh, *Bioinformatics*, 2003.

Kleinberg, 1999*

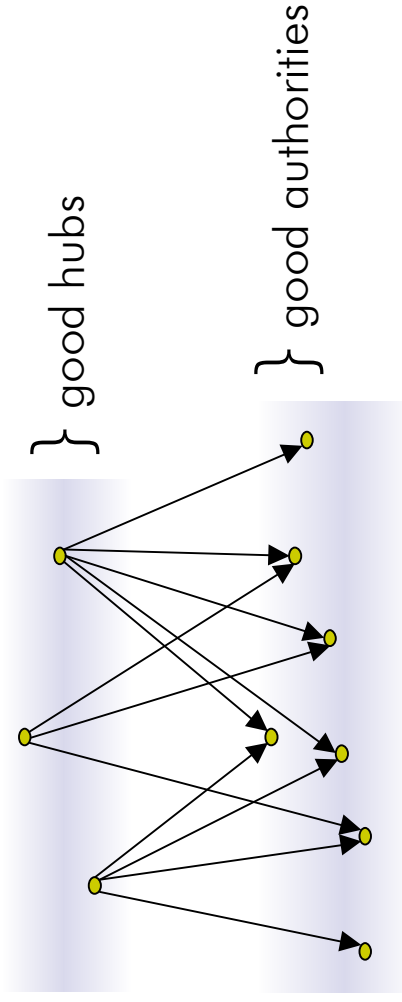
- Motivated by demands of web searching
- Step 1: Use text-based search methods to identify a candidate graph containing relevant websites and their neighbors.



*Kleinberg, J.M. Authoritative sources in a hyperlinked environment. *Journal of the ACM*. 1999.

Kleinberg, 1999

- Relevant search results might be:
 - Hubs – pages which *point to* many good authorities
 - Authorities – pages which are *pointed to* by many good hubs

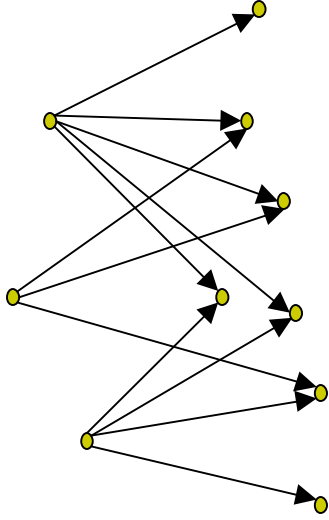


- Step 2: Compute hub and authority scores for every node in the candidate graph.

Kleinberg, 1999

- Denote:
 - $x_{1p}(k)$ = hub score of node p at iteration k
 - $x_{2p}(k)$ = authority score of node p at iteration k

- Update rule:



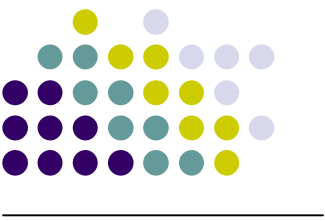
$$x_{2p}(k+1) = \sum_{q:(q,p) \in E} x_{1q}(k)$$

i.e. the sum of hub scores of nodes that point to node p

$$x_{1p}(k+1) = \sum_{q:(p,q) \in E} x_{2q}(k)$$

i.e. the sum of authority scores of nodes that are pointed to by node p

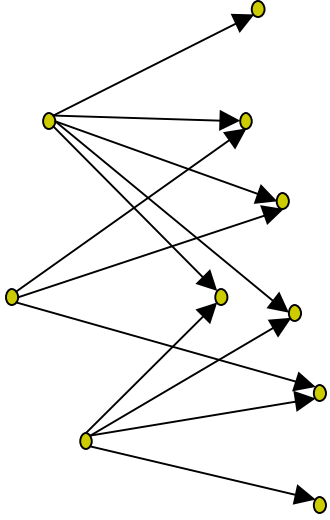
- Normalize the scores so that $\sum_p x_{ip} = 1$ and repeat.



Kleinberg, 1999

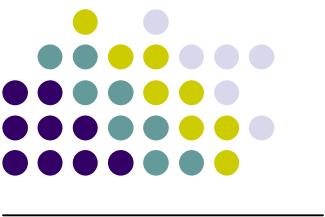
- Denote:
 - $x_{1p}(k)$ = hub score of node p at iteration k
 - $x_{2p}(k)$ = authority score of node p at iteration k

- Update rule:



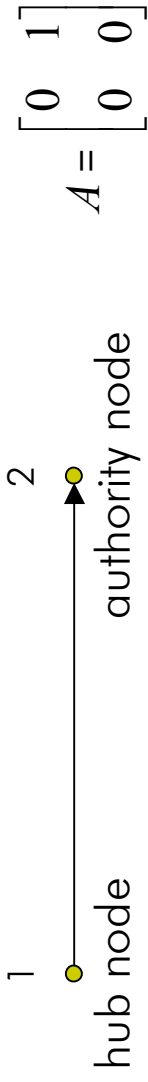
- Stack the scores $x_{1p}(k)$ into a vector $[x_1]_k$, then stack $[x_1]_k$ and $[x_2]_k$.
- Let B be the node-node adjacency matrix of the candidate graph. Then:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & B \\ B' & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k$$



Blondel, Van Dooren, et al., 2004*

- Views Kleinberg's iteration as a comparison between the web graph and a *hub-authority* graph:



- Observe that the matrix form of Kleinberg's update can be written as follows:
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & B \\ B' & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k = (A \otimes B + A' \otimes B') \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k$$
- Is this generalizable to any two graphs G_A and G_B ? YES.

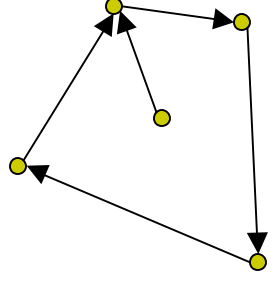
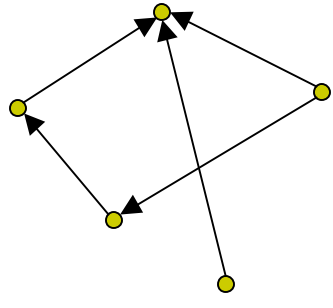
*Blondel, V., Gajardo, A., Heymans, M., Senellart, P., Van Dooren, P. A measure of similarity between graph vertices: applications to synonym extraction and web searching. *SIAM Review*, v. 46(4), 647-666. 2004.

Coupled edge and node scoring

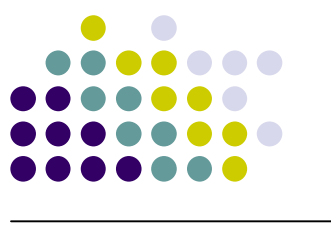
- Idea: use this iterative approach to assign edge similarity scores as well as node similarity scores.
- Couple the definitions in the following manner:

x_{ij} = similarity between node i in G_B and node j in G_A
= sum of pairwise similarities between adjacent edges

Y_{ij} = similarity between edge i in G_B and edge j in G_A .
= sum of similarities of source and terminal nodes



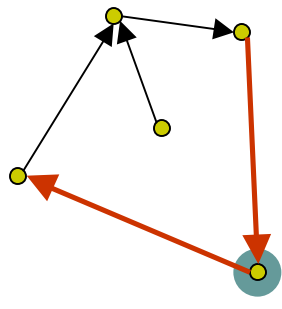
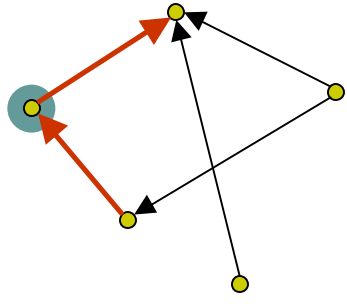
Coupled edge and node scoring



- Idea: use this iterative approach to assign edge similarity scores as well as node similarity scores.
- Couple the definitions in the following manner:

x_{ij}  = similarity between node i in G_B and node j in G_A
= sum of pairwise similarities between adjacent edges

Y_{ij} = similarity between edge i in G_B and edge j in G_A .
= sum of similarities of source and terminal nodes

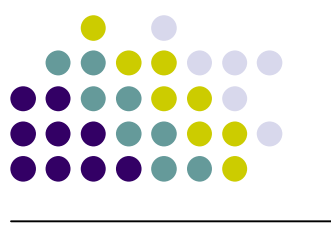
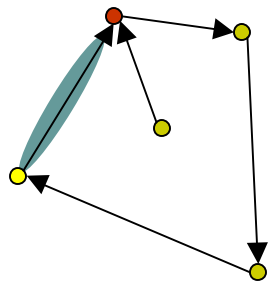
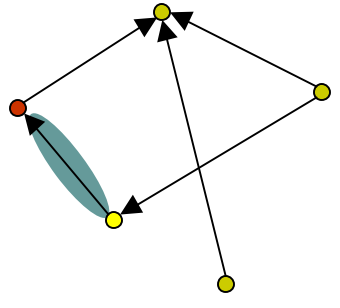


Coupled edge and node scoring

- Idea: use this iterative approach to assign edge similarity scores as well as node similarity scores.
- Couple the definitions in the following manner:

x_{ij} = similarity between node i in G_B and node j in G_A
= sum of pairwise similarities between adjacent edges

y_{ij} = similarity between edge i in G_B and edge j in G_A .
= sum of similarities of source and terminal nodes



Coupled edge and node scoring

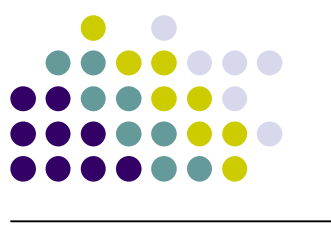
- Idea: use this iterative approach to assign edge similarity scores as well as node similarity scores.
- Couple the definitions in the following manner:

x_{ij} = similarity between node i in G_B and node j in G_A
= sum of pairwise similarities between adjacent edges

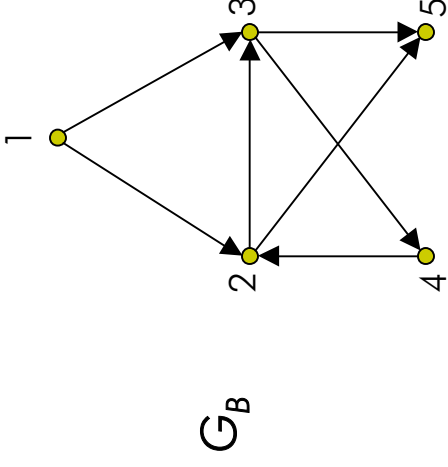
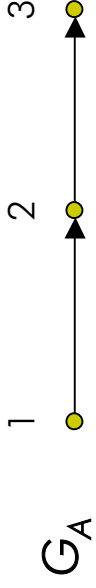
Y_{ij} = similarity between edge i in G_B and edge j in G_A .
= sum of similarities of source and terminal nodes

$$\bar{x}_{k+1} = [A_S \otimes B_S + A_T \otimes B_T] \bar{y}_k$$
$$\bar{y}_{k+1} = [A'_S \otimes B'_S + A'_T \otimes B'_T] \bar{x}_k$$

$$[A_S]_{ij} = \begin{cases} 1 & s(j) = i \\ 0 & \text{else} \end{cases} \quad [A_T]_{ij} = \begin{cases} 1 & t(j) = i \\ 0 & \text{else} \end{cases}$$



Example

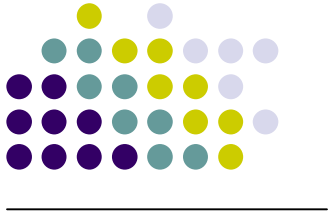


Blondel, Van Dooren, et al.
similarity scores

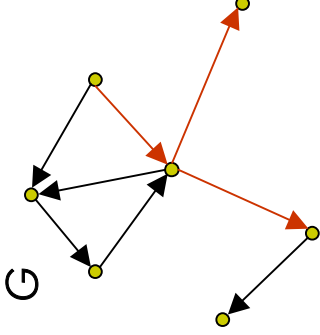
nodes	1	2	3
1	0.443	0.104	0
2	0.280	0.396	0.086
3	0.086	0.396	0.280
4	0.222	0.049	0.222
5	0	0.104	0.443

Coupled model
similarity scores

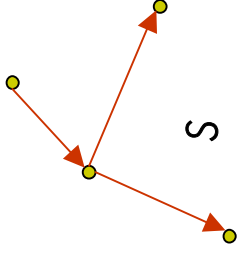
nodes	1	2	3
1	0.324	0.054	0
2	0.177	0.587	0.018
3	0.018	0.587	0.177
4	0.127	0.010	0.127
5	0	0.054	0.324



Application: Graph Matching



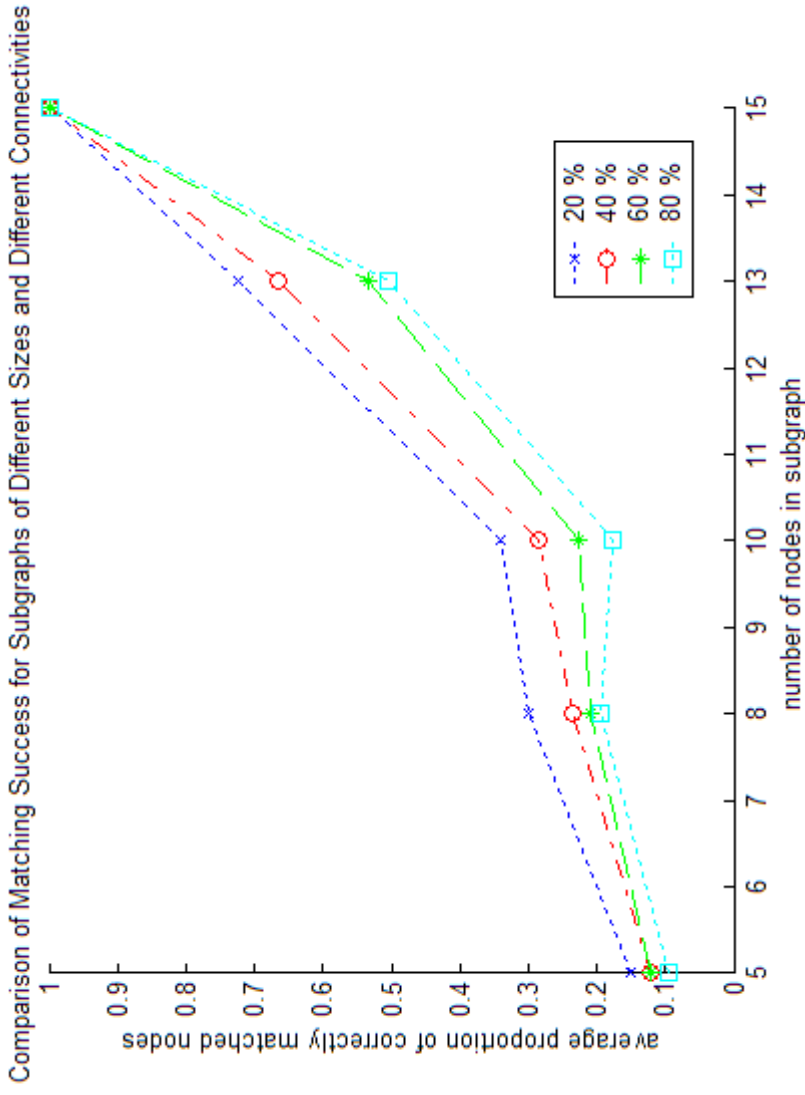
- Task: subgraph matching
 - Generate a random graph, G
 - Select a subgraph, S
 - Compute the node similarity matrices between G and S
 - Apply the Hungarian algorithm to `best' match the nodes of S to those in G by finding a matching that maximizes the sum of matched scores.
 - Record successes for nodes that are matched with their original identifier



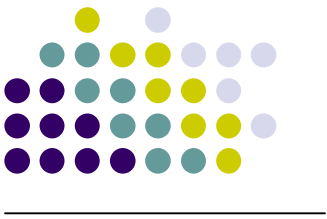
Yields a lower bound on the success of the matching process

Application: Graph Matching

- Some preliminary performance results on identifying subgraphs



Current and future work



- For a self-similarity score matrix, is an isomorphic mapping always among the maximum weight matchings?
- Is there an easily computable way to distinguish non-isomorphic graphs?
- What can be inferred about a pair of graphs from a similarity measurement?
- What kinds of tasks is this measure appropriate for?



Acknowledgments

- George Verghese, MIT
- Paul Van Dooren, Université catholique de Louvain

Work supported by a NSF Graduate Research Fellowship.